TESTJAHRBUCH 1988 C16

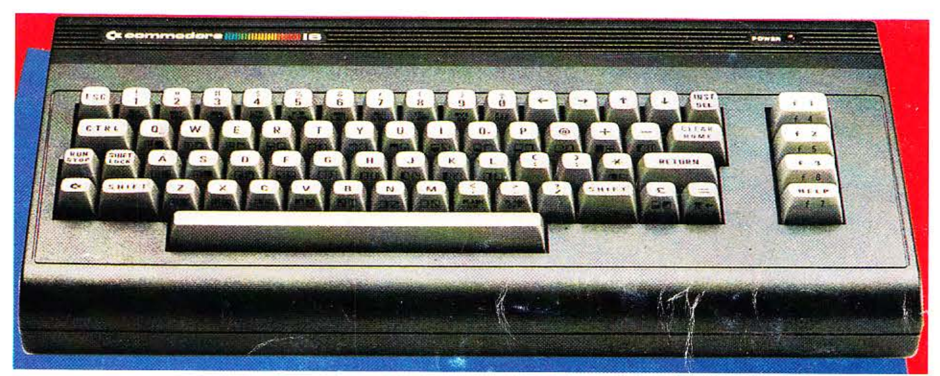
COMMODORE WELT

SPECIAL

P/4 C 16 116

TEST-JAHR-BUCH 1988







Tests
Tips
Tricks
Kaufberatung

Alles über Ihren C16/P4

Das Handbuch für C16/P4



COMMODOREWELT-SPECIAL 3/88

FÜR SIE GETESTET

FÜR SIE GESCHRIEBEN



COMMODOREWELT-SPECIAL 3/88



Die „Black Beauties" von Commodore können mehr, als Viele vermuten. Dieses Jahrbuch zeigt es Ihnen

FÜR SIE AUSPROBIERT: TIPS & TRICKS

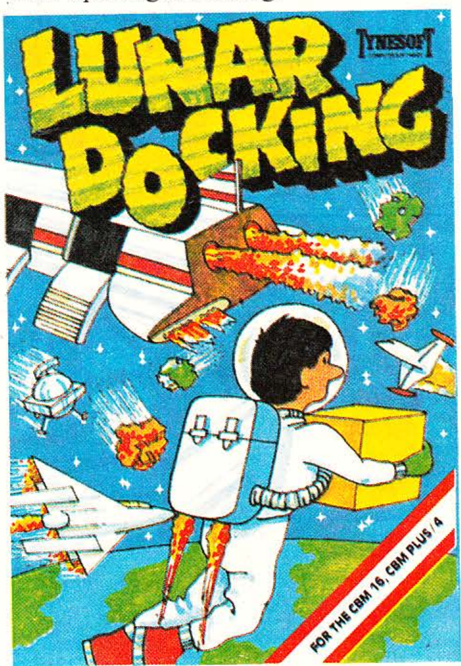
SPIELE SATT

Natürlich können die schwarzen Rechner von Commodore dem ehrwürdigen C 64 in punkto Spiele noch nichts vormachen. Doch was mittlerweile angeboten wird, ist meist auch nicht von schlechten Eltern. Wir untersuchten diverse Spiele aus dem Angebot der Firma Stechmann.

LUNAR DOCKING

Hersteller: Tynesoft

Ziel des Spieles ist es, eine Raumstation mit neuer Energie zu versorgen. Hierfür steht als Spielfigur ein Astronaut mit Raumanzug bereit, der die wichtige Energieversorgung übernehmen soll. Von einem Spaceshuttle aus werden die einzelnen Elemente auf der Erdoberfläche verstreut. Diese müssen nun aufgesammelt und in den Frachtraum der Raumstation gebracht werden.
Schwierigkeiten hierbei:
1. Es kann nur jeweils eine Ration mitgeführt werden. Bei mehr gibt es eine Spielfigur Abzug.



2. Über der Erde kreisen jede Menge Raumschrott und Satelliten. Diese sind bei Berührung ebenfalls tödlich. Daher muß ein vernünftiger Weg durch die Unordnung gefunden werden, dieser Spielteil ähnelt ein bißchen dem altbekannten Frogger.
3. Um die Ladung erfolgreich zu verstauen, muß sehr vorsichtig manövriert werden. Bei Unaufmerksamkeiten wird dies als Versuch gewertet, eine zweite Ladung annehmen zu wollen. Effekt: Spielerverlust!
Beurteilung: Empfehlenswert.
Eine sinnvolle Spielaufgabe ohne mörderische Ballerei, die auch nach längerer Spielzeit nicht langweilig wird. Da es sehr viel Ladung nach oben zu bringen gilt, dürfte die Aufgabe auch nach einiger Übung nicht all zu leicht zu erfüllen sein.

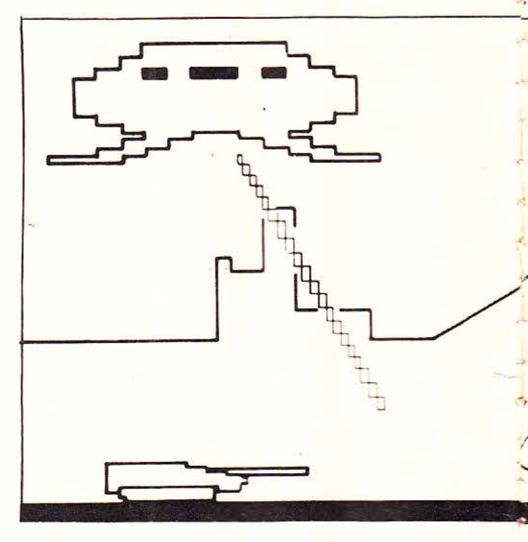
JACK ATTACK

Hersteller: Commodore

Dieses Modul gehört zur Kategorie der Original-Commodore-Spiele, deren Qualität ja noch nie besonders anspruchsvoll war. Außerdem muß fairerweise erwähnt werden, daß dieses Spiel bereits seit 1983 existiert, also nicht mehr zu den jüngsten gehört. Doch diese Meinung ist natürlich subjektiv, denn nicht jeder ist verwöhnt wie wir, was Spiele anbelangt. Bei Jack Attack geht es darum, mit der Spielfigur verschiedene Plattformen so zu verschieben, daß umherschwebende Ballons zerquetscht werden. Die Grafik ist hierbei recht primitiv gehalten. Die beiden einzigen Gefahren bei diesem Spiel liegen darin, entweder von den Ballons erwischt oder



unter einer verschobenen Plattform begraben zu werden.

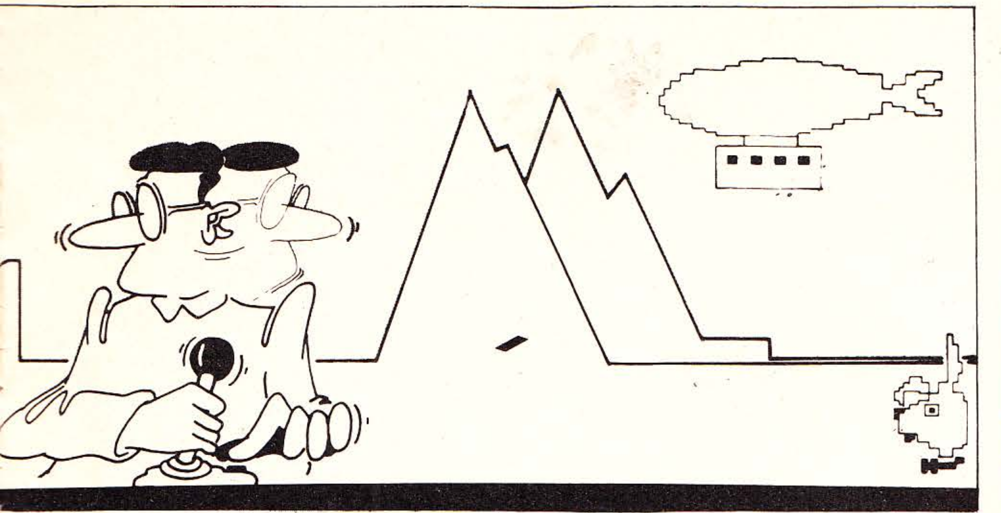


Die Plattformen können sowohl gezogen als auch geschoben werden, sollten allerdings mehrere Steine aufeinanderliegen, so ist die Gefahr groß, eine Plattform auf die Spielfigur zu bekommen, woraufhin diese natürlich dahinscheidet.
Beurteilung: Für Kinder ist dieses Spiel durchaus empfehlenswert, allerdings sollte klar sein, daß es natürlich in keiner Weise mit den neueren Spielen, welche für diese Rechner erschienen sind, konkurrieren kann.

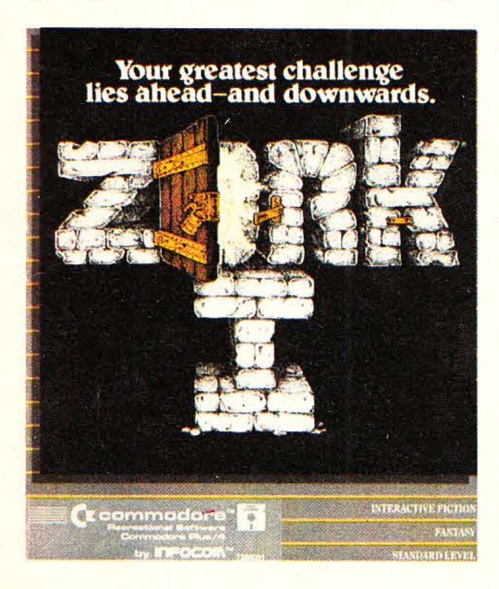
ZORK

Hersteller: Infocom

Den älteren unserer Leser dürfte Infocoms "Zork" wohlbekannt sein. Mittlerweile gibt es drei Versionen dieses Adventures, welche seit neuestem auch alle drei für den Plus/4 bzw. C 16/116 mit 64 KByte erhältlich sind! Damit ist eine weitere Domäne des C 64 gebrochen, denn anspruchsvollere Adventures als die Infocoms dürften schwerlich zu finden sein. Die Ausstattung des Paketes ist, wie gewohnt, vorzüglich gestaltet. So findet der Käufer in seiner Verpackung eine mysteriöse Karte des alten Königreiches, außerdem eine Art Geschichtsbuch, mit dessen Hilfe das Spielgeschehen hautnah vor Augen geführt wird. Hier natürlich gleich vorweg: Zork ist auch in der 3.5er-Version ein englischsprachiges Adventure, daher sollten wirklich nur solche Leute es erwerben, die dieser Sprache auch halbwegs mächtig sind. Gleichermaßen stellt das Adventure dadurch auch eine gute Möglichkeit dar, sich mit dieser Sprache vertraut zu machen.
Weiterhin sind einige sehr nützliche Hinweise in einer Art "User



Guide" aufgestellt, die natürlich nicht all zu viel über das Spielgeschehen informieren.
Inhalt von Zork: Irgendwo im fernen "Underground Empire" sollen sagenhafte Schätze und Geheimnisse verborgen sein. Diese gilt es nun zu finden. Wie sollte es anders sein, auf dem Weg zu diesen Schätzen lauern natürlich die vielfältigsten Aufgaben und Gefahren auf den Spieler. Das besondere an Infocoms Adventure ist dessen gewaltiger Sprachschatz. So sind die Mög-



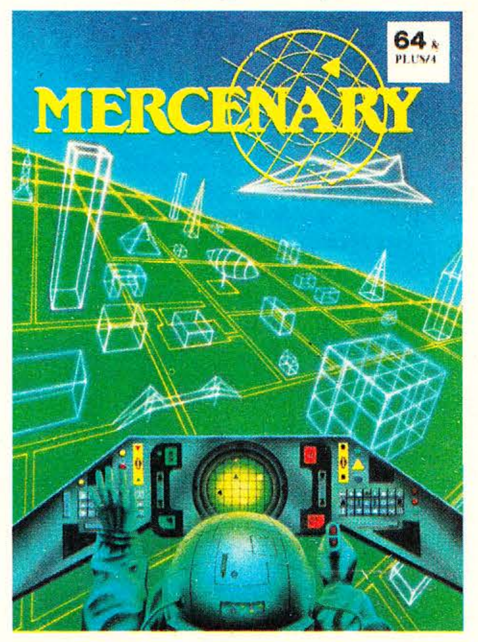
lichkeiten, um Aktionen durchzuführen, sehr umfangreich. Auch die jeweiligen Ortsbeschreibungen sind sehr ausführlich gehalten. Bei diesem Umfang ist es klar, daß nicht mehr viel Speicherplatz für Grafiken übrigbleiben konnte. Daher muß der Spieler, wie bei allen Infocom-Adventures, auf Grafiken verzichten. Alle Informationen müssen daher aus dem Text gezogen werden, ein Grund mehr für gute Englisch-Kenntnisse.

Beurteilung: Sehr empfehlenswert! Diese brillante Aufmachung des Spieles untermauert einmal mehr den Ruf Infocoms als Rolls-Royce unter den Adventureanbietern. Durch die hervorragenden Begleitmaterialien lohnt es sich wirklich, dieses Spiel anzuschaffen.
Die Besitzer eines Plus/4 oder erweiterten 16ers müssen nun nicht mehr auf Adventure-Genüsse der Extraklasse verzichten.

MERCENARY

Hersteller: Novagen Software

Auch bei diesem Spiel handelt es sich um einen alten (C 64-)Bekannten. Die Adaption auf Plus/4 oder erweiterte 16er steht dem Original jedoch in nichts nach und stellt unter Beweis, daß die Fähigkeit des „Einsteigercomputers" denen des C 64 nicht nachstehen müssen. Neu ist außerdem, daß es für die Plus/4 wie die 64er-Version nun die Fortsetzung des Kassenschlagers gibt: Mercenary – The second City. Diese ist für all diejenigen gedacht, welche das Unmögliche möglich machten und die Mercenary-Aufgabe bereits gelöst haben. Die Aufgabe braucht eigentlich nicht mehr lange erklärt werden, denn das Programm ist, wie gesagt, bereits hinlänglich bekannt. Für die Neulinge ein "Shortcut":
Mit Ihrem Raumschiff sind Sie auf einer fremden Welt notgelandet, dem Planeten Targ. Zu allem Überfluß herrscht auf diesem Planeten auch noch Krieg zwischen zwei Mächten, den Sie neben dem eigentlichen Spielzweck natürlich auch noch überleben müssen. Einziger Gefährte bei dieser Aufgabe ist der Bordcomputer Benson, der öfters seine Meinung von sich gibt und meist die einzige Hilfestellung zur Lösung der Aufgabe ist. Ach ja, die Aufgabe: Suchen Sie nach genügend Geld und Ersatzteilen, um Ihr Raumschiffwrack wieder flugtüchtig zu machen. Damit der arme Held nicht zu Fuß durch das Abenteuer laufen muß, stehen verschiedene Fahr- und Flugzeuge zur Fortbewegung bereit, die aller-



dings natürlich auch Geld kosten. Geld können Sie sich in einigen Jobs verdienen, die von Zeit zu Zeit angeboten werden. Soviel das Altbekannte.
Second City geht auf dieselbe Weise weiter, nur fehlen Ihnen hierbei die nützlichen Hinweise des Handbuchs, Sie sind also ganz auf sich alleine gestellt.
Beurteilung: Sehr empfehlenswert! Diese Version hat nichts vom Spielreiz des 64er-Vorbildes eingebüßt und überzeugt auch auf dem Plus/4 durch gute Animation und großen Spielreiz.

SABOTEUR

Hersteller: Durell Software

Der Titel läßt Schlimmes vermuten, doch es kommt noch ärger.
Ein Spiel, welches wohl nur Personen mit starken "Rambo"-Ambitionen empfohlen werden kann, die irgendwelche Gelüste am Steuerknüppel ausleben wollen. Hierfür steht eben dieser „Saboteur von Chaos", so der Originaltitel, zur Verfügung. Mit einem vermummten (Herr Zimmermann wird sich freuen) Männchen muß versucht werden, in einem als Warenhaus getarnten Rebellennest eine Diskette

zu finden. Auf dieser Diskette befinden sich irgendwelche Namen irgendwelcher Untergrundkämpfer, die irgendwer benötigt und deswegen Sie beauftragt hat, diese zu holen. So viel konnten wir der Anleitung entnehmen, obwohl diese durch die Übersetzung ins deutsche doch etwas konfus geriet. Kleiner Auszug gefällig?

„Wenn Sie die WERF-GEBRAUCH Steuerung benutzen, wenn ein Gegenstand in Ihrem NEAR DISPLAY ist, dann wird der NEAR Gegenstand ein HELD Gegenstand und wenn Sie zu der Zeit einen HELD Gegenstand hatten, wird daraus ein NEAR Gegenstand."

Alles klar? Wenn nicht, machen Sie sich nichts daraus, denn das Spiel funktioniert auch ohne solch tiefgründiges Wissen, Hauptsache, Sie machen möglichst viele Wächter nieder. Zur Abwechslung können Sie umherlaufende Hunde erledigen, das bringt jedoch keine Punkte. Es gibt sogar Unterschiede zwischen Killen und Killen: „Wächter mit Waffe getötet: 100 Punkte – Wächter durch Schlagen oder Treten getötet: 500(!) Punkte." (Richtiger NINJA-Stoff). So die Anleitung: Wer nach alledem immer noch Verlangen nach solch einem Spiel hat, kommt wohl auch für das folgende in Frage.

Beurteilung: Nicht empfehlenswert. Eines weiteren Kommentares enthalten wir uns.

COMMANDO

Hersteller: Elite

Hierbei handelt es sich um ein ebenso fragwürdiges Spielchen wie das vorhergehende. Die Story ist aus anderen ähnlich dumpfen Spielen bereits bekannt, hier nur das Wichtigste: Die Spielfigur (sinnigerweise „Superjoe" genannt) ist der letzte Retter für die zivilisierte Menschheit und darf sich dementsprechend auf dem Bildschirm austoben. Oberstes Gebot: Du darfst kein Mitleid zeigen (Zitat Originalanleitung). Damit das ganze etwas einfacher gerät, führt unser Joe ein M60-Maschinengewehr (was auch immer das sein mag) mit sich und muß dieses natürlich dazu benutzen, reihenweise die Feinde niederzumähen.

Beurteilung: Siehe „Saboteur".

BOMB JACK

Hersteller: Elite

Bomb Jack kennen die meisten wahrscheinlich bereits vom Spielautomaten her, auf welchem dieses Männchen umherturnte. Nun dürfen auch die C 16-Besitzer in den Genuß eines heimischen Jack's kommen, nachdem ja die 64er-Version seit geraumer Zeit bereits vorhanden ist.

Da es sich hier um die Version für 64 K handelt, wurde die Originalgrafik etwas abgespeckt, der Sound fiel ebenfalls sehr mager aus. Doch trotzdem wurde für diese Verhältnisse Erstaunliches geleistet, man möchte kaum glauben, was doch mit einem unerweiterten 16er bereits realisierbar ist. Die Grafik entspricht noch weitgehend dem Original, auch wurde eine sehr gute Auflösung erreicht. Ziel des Spieles ist es, sich mit der Spielfigur Jack über verschiedene Plattformen zu bewegen und dabei



Bomben aufzusammeln. Dabei stören natürlich etliche umherschwirrende Viecher, denen es auszuweichen gilt. Jack kann per Feuerknopf springen, wird dieser Knopf mehrmals hintereinander betätigt, so schwebt er ganz langsam nach unten. Von Zeit zu Zeit tauchen Extra-Punkte auf, die es zu fangen gilt. Hierfür gibt es zum Beispiel eine Multiplizierung des Punktekontos oder auch die – leider nur kurzzeitige – Unsterblichkeit Jacks. Während dieser kann Jack nun seinerseits Jagd auf all das gräusliche Getier machen, welches ihm im Normalfall das Leben schwer macht.

Beurteilung: Empfehlenswert. Hier wurde wirklich eine ordentliche Fassung des bekannten Arcade-Games verwirklicht, die natürlich aufgrund des Speichervolumens nicht ganz so aufwendig gehalten werden konnte wie das Original. Trotzdem ist Bombjack ein Spiel, welches so schnell nicht langweilig werden dürfte.

EUROPEAN GAMES

Hersteller: Tynesoft

Auf der inzwischen schon leicht abgeflauten Welle der Sport-Spiele kommt Tynesoft mit den Europa-Meisterschaften angeschwommen. Diese wird im Falle der C16-Version in fünf Disziplinen ausgetragen:

- Gewichtheben
- Rudern
- Weitsprung
- Hammerwurf
- Schwimmen.

Alle fünf Disziplinen lassen sich mit einem Wort umfassen: Ein harter Tag für den Joystick. Denn im Gegensatz zu den bekannten Sport-Spielen, in denen es auf gute Koordination der Bewegungsabläufe ankommt, reicht hier Geschwindigkeit. So muß bei jeder Disziplin der arme Knüppel mit geradezu aberwitziger Geschwindigkeit von einer Seite zur anderen gedroschen werden, um der jeweiligen Spielfigur etwas Tempo einzuhauchen. Sollte sich der arme Spieler im Besitz eines Uralt-Joysticks befinden, der noch keine Mikroschalter besitzt, so dürften diese Übungen denen in einem Body-Building-Studio gleichkommen. Wir zogen zum Test den Quickshot II von SVI heran, der einerseits mit Mikroschaltern ausgestattet ist und andererseits über eine enorme Länge verfügt, so daß mit geringstem Kraftaufwand der Rechts-Links-Wechsel durchgeführt werden kann. Trotz alledem rauchten die Köpfe, als versucht wurde, den jeweiligen Highscore zu überbieten. Erfreuliches wurde hier beim Spielverlauf verwirklicht: Nach jedem Durchgang fragt der Computer, ob noch einmal gespielt werden soll. Erst wenn diese Frage verneint wird, geht es weiter zur nächsten Disziplin. So kann jeder in Ruhe üben, um den richtigen Bogen zu finden. Die Grafikanimation darf als mittelprächtig bezeichnet werden, diese ist aber durchaus ausreichend.

Bewertung: Bedingt empfehlenswert. Zwar sind die Disziplinen im einzelnen recht schön gestaltet, doch ist durch die vergleichsweise primitive Steuerung viel Reiz aus diesem Spiel genommen.

Und nun genug gespielt, denn Kingsoft wartet mit diversen Leckerbissen für die Besitzer der C 16-Reihe auf: Arbeits- und Anwendungsprogramme, die sich hinter denen erheblich teurerer Geräte nicht zu verstecken brauchen.

MUSIC MASTER

Etwas skeptisch waren wir durchaus, als es ans Ausprobieren dieses Programmes ging. Stand doch auf der Verpackung „verwandelt Ihren Computer in einen Sequenzer und Synthesizer". Große Worte, bedenkt man das beschränkte Speichervolumen der 16er, auf denen dieses Programm läuft. Doch die Wirklichkeit holte uns schnell ein, es stellt sich heraus, daß nicht zuviel versprochen wurde!
Auf dem Bildschirm erscheinen zunächst eine Klaviertastatur sowie die einzelnen Einsteller für besondere Effekte. Auf der Klaviatur sind die Tasten des Computers verzeichnet, mit welchen die einzelnen Töne erzeugt werden können. Weiterhin erscheint jeder Taste zugeordnet die passende Note. So sind schon nach kurzer Einübungszeit zusammenhängende Musikstücke möglich. Nun folgt die eigentliche Bedienung des Programmes, denn es sind wirklich hervorragende Features enthalten. Hiervon zeugt alleine schon die äußerst umfassende Anleitung, die wirklich jeden Programmpunkt gut zu erklären weiß.
Mit dem Programm sind sehr viele Spezialeffekte möglich. So kann zum Beispiel ein Musikstück editiert werden. Das heißt, daß je nach Können des Anwenders, mehr oder weniger schnell, Noten eingegeben werden, welche sich der Computer merkt. Diese können dann auf vielfältige Weise verändert und bearbeitet werden. Es ist möglich, einzelnen, schon gesetzten Noten neue Oktaven zuzuordnen, neue Spieldauern zu bewirken, Tempo zu verändern und vieles mehr. Auch kann auf zwei Kanälen hintereinander aufgenommen werden, die wahlweise synchron geschaltet werden und so zu einem harmonischen Ganzen zusammengefügt werden. Weiterhin ist es möglich, einzelne Musikstücke, die selbst erstellt wurden, auf Diskette oder Kassette abzuspeichern! Kingsoft geht sogar weiter und bietet in der Anleitung auch noch Programmlistings an, die die Übernahme von Melodien in andere Programme er-



MUSIC MASTER

- **Verwandelt Ihren Computer in einen Sequenzer und Synthesizer**

PAINTBOX

- **Mal- und Zeichenprogramm in 121 Farben**

COMMODORE 16 116 PLUS/4

möglichen, wenn diese mit Music Master erstellt wurden.
Weiterhin sind im Programm verschiedene Rhythmen gespeichert, die zusätzlich verwendet werden können, um beim Editieren noch höhere Qualität zu erreichen. Selbstverständlich werden diese Rhythmen auch übernommen, wenn die Melodie abgespeichert oder eingeladen wird.

Fazit:

Hier konnten nur die wichtigsten Programm-Finessen aufgeführt werden, da der Testbericht sonst erheblich länger geworden wäre. Dem Käufer werden wohl auf längere Sicht noch viele kleine Einzelheiten auffallen, die die Arbeit mit Music Master zur wahren Freude geraten lassen. Es ist schon erstaunlich, was der Programmierer mit diesem Synthesizer aus dem C 16 herausgekitzelt hat.

PAINTBOX

Da wir gerade beim Kreativen sind: Auch ein Malprogramm bietet Kingsoft mit der Paintbox an. Dieses Malprogramm bietet alle Möglichkeiten, die bereits von professionellen Programmen her bekannt sind. So sind neben den Grundoperationen wie Kreise zeichnen, Linien, Strahlen und Ähnliches erzeugen natürlich auch Fill-, Lösch- und Farbwahlroutinen vorhanden. Alleine zum Zeichnen stehen nicht weniger als sieben Möglichkeiten zur Verfügung.
Weiterhin kann der Anwender aus einer Farbpalette bis zu 121 Farb-Variationen auf dem Bildschirm erzeugen. Es ist natürlich auch die Möglichkeit gegeben, mit verschiedenen Pinselstärken zu arbeiten, hierfür stehen 8 verschiedene Varianten zur Auswahl.
Neue Features birgt das Programm indes auch. So ist es beispielsweise möglich, die Geschwindigkeit des Cursors zu verändern, mit dem die Grafiken erzeugt werden. Dies geschieht, ähnlich zum Amiga, über ein Untermenü, in welchem an einem Balkendiagramm die Geschwindigkeit variiert werden kann.
Auch ist es möglich, die Helligkeit der einzelnen Farben noch zu verändern, so daß dieses Programm eine wirklich erstaunliche Farbvielfalt erzeugen kann. Um die erstellten Bilder zu sichern beziehungsweise bereits gespeicherte Bilder neu zu bearbeiten, sind Lade- und Save-Routinen vorhanden, die sowohl für Diskette als auch für Kassette verwendet werden können.

Fazit:

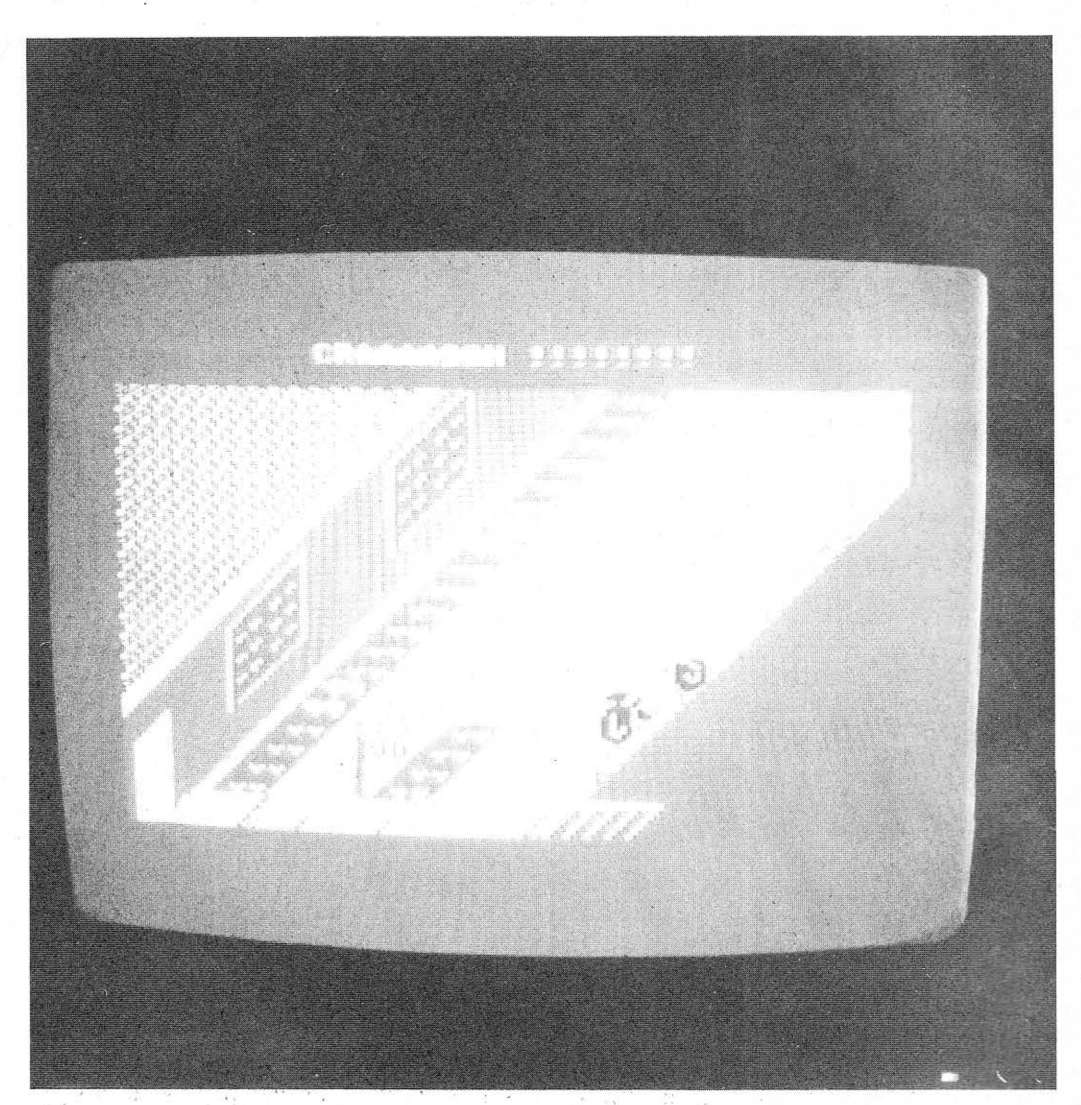
Neben dem umfangreichen Befehlssatz glänzt das Programm durch eine außerordentlich hohe Geschwindigkeit (sofern dies vom Anwender gewünscht wird). Auch die Sonderfunktionen konnten durchaus überzeugen. Die ausführliche deutsche Anleitung rundet das positive Urteil von Paintbox ab und verdeutlicht, das der C 16 zu Großem fähig ist.

Alles in allem bliebe zu sagen, daß sich das Angebot qualitativ hochwertiger Software für die C 16/116/P4 hervorragend entwickelt hat.
Es bleibt zu hoffen, daß Kingsoft und dem Oscar-Autor Udo Gertz (der mit der Winterolympiade) auch weiterhin so erstaunliche Programme einfallen. Weiterhin im Angebot von Kingsoft sind außerdem noch Business-Programme zur Datenverwaltung und Kalkulation mit den 16er. Diese werden derzeit noch einem ausführlichen Test unterzogen, mehr darüber in der Normal-Ausgabe der COMMODORE-WELT.

T. Seibt

QUIWI

Dieses Spiel dürfte den meisten bereits bekannt sein. Kingsoft vertreibt das Ganze seit neuestem auch für den C 16. Der Spielinhalt ist schnell erklärt, was allerdings nichts über die Qualität aussagt. Quiwi ist ein Ratespiel, das bis zu zwölf(!) Personen spielen können. Einsamkeit verursacht das Game also kaum.
Und nun zum eigentlichen Spiel: Der Käufer erhält eine Diskette, deren beide Seiten jeweils randvoll gepackt sind. Dies gibt schon Auskunft über den Umfang des Spieles. Exakt 3960 Fragen warten auf die Spieler.
Nach dem Laden des Programmes fragt das Programm ab, ob die richtige Diskettenseite eingelegt ist. Dies ist wichtig, wenn Sie alle Fragen einer Seite beantwortet haben oder auswendig kennen. Bei rund 2000 Fragen pro Diskettenseite dürfte dies eine Weile dauern.
Nach dem Einlesen gelangen Sie in das Menü zur Namenseingabe. Wie gesagt, sind hier bis zu zwölf Personen möglich. Nach dieser Prozedur geht das eigentliche Spiel los. Quiwi, der Spielheld, steht auf einer Wiese und spannt den Bogen. Mit der Feuertaste des Joysticks können Sie den Pfeil auf eine laufende Zielscheibe abfeuern, deren Farben verschiedene Wissensgebiete markieren. Insgesamt sind sechs Gebiete vorhanden: Verschiedenes, Erdkunde, Geschichte/Politik, Sport/Humor, Kunst/Literatur und Wissenschaft/Technik. Mit etwas Übung und Geschicklichkeit hat man bald heraus, wie die einzelnen Farben getroffen werden. Ziel des Spieles ist es, in jedem Wissensgebiet die vorgegebene Anzahl von richtigen Antworten zu geben (diese Anzahl läßt sich von eins bis drei bestimmen). Der Spieler gelangt nun in das Menü, in welchem die Punktestände verzeichnet sind. Das betreffende Wissensgebiet wird nachgeladen, der Spieler erhält seine Frage. Nun muß diese Frage richtig beantwortet werden, dann darf man sich die Lösung zeigen lassen. Nun kann man (nicht schummeln!) angeben, ob die eigene Antwort richtig oder falsch war, oder ob keine Wertung erfolgen soll. Keine Wertung gilt zum Beispiel, wenn sich die Mitspieler uneinig sind, ob die Antwort richtig oder falsch war. Wurde die richtige Antwort gegeben, erhält der Spieler einen Punkt unter dem betreffenden Wissensgebiet. In dieser Weise geht es nun weiter, bis der Spieler eine Frage falsch beantwortet. Dann ist der nächste Spieler an der Reihe. Wenn nun ein Spieler alle Wissensgebiete mit ausreichend richtigen Antworten durchgearbeitet hat, gelangt er in die Endrunde. Nun dürfen die Mitspieler das Wissensgebiet bestimmen, für das der Endrundenspieler antworten muß. In der Endrunde muß der Spieler noch einmal drei Fragen richtig beantworten. Antwortet er falsch, wird eine richtige Antwort wieder abgezogen.
Quiwi läßt sich wohl am besten mit dem Spiel „Trivial Pursuit“ vergleichen, das im vergangenen Jahr wahre Verkaufsrekorde erreicht hat. Es ist also sehr wahrscheinlich, daß diese Computervariante des Fragenspieles sehr viele begeisterte Käufer finden wird.
Übrigens: Wer tatsächlich alle Fragen durch haben sollte (falls dies überhaupt gelingt), für den bietet Kingsoft noch einen zusätzlichen Service an. Es können in naher Zukunft weitere Datendisketten geordert werden, die wieder voll mit Fragen sein werden.



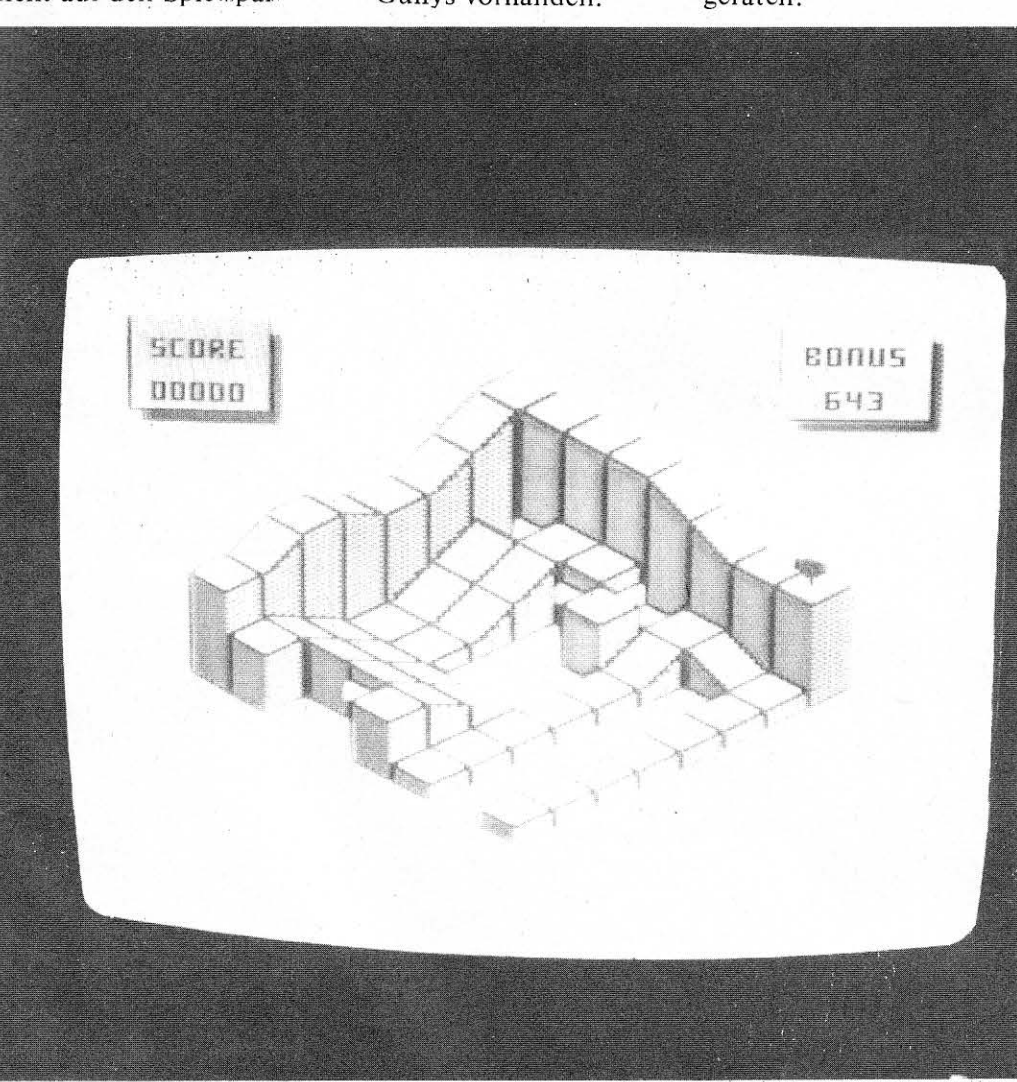
PAPER BOY

Kingsoft hat für Elite die Umarbeitung dieses Arcade-Games übernommen. Vielen dürfte dieses Spiel beispielsweise vom 64er bekannt sein, jetzt ist jedenfalls die 16er-Version fertig. Das Thema ist bekannt, es geht um das

allmorgendliche Zeitungsaustragen mit dem Fahrrad. Es ist erstaunlich, daß trotz des Speichermankos des C 16 eine sehr passable Umsetzung der Grafik auf diesen Rechner geschehen ist. Es kommt der 64er-Version schon ziemlich nahe. Daß bei dieser Grafikleistung der Sound etwas leiden mußte, ist klar; dies wirkt sich allerdings nicht auf den Spielspaß aus. Der Sinn des Spieles dürfte bekannt sein: Es geht darum, in einer Kleinstadt per Fahrrad Zeitungen auszutragen. Mit dem Joystick lenkt man das Gefährt durch die Straßen, per Feuerknopf wirft der Paperboy eine Zeitung. Nun muß gut gezielt werden, denn jeder Wurf muß sitzen. Der Paperboy hat neun Zeitungen. Wirft er eine an ein Zeitungsrohr, so gibt es Bonuspunkte. Trifft er das offene Fenster oder die Stufe vor der Haustür, gibt es jeweils die maximale Punktzahl. Wird daneben geworfen, geht die Zeitung verloren und es gibt keine Punkte. In unregelmäßigen Abständen tauchen Gefahren für den Radler auf:

- Auf der Straße sind Gullys vorhanden. Diese zu überfahren, führt zum CRASH.
- Auf den Straßen und Gehwegen liegt allerlei Gerümpel herum. Dieses sollte nach Möglichkeit umfahren werden, um nicht zu stürzen.
- Ab und zu kommen bewegliche Hindernisse, zum Beispiel herrenlose Kinderwagen, einhergefahren. Diesen muß natürlich ausgewichen werden.

Jeder CRASH fordert ein Radler-Leben, drei davon stehen pro Spiel zur Verfügung.
Für einen Preis von 19 Mark kann dieses Spiel den 16er-Besitzern durchaus empfohlen werden, die Umsetzung ist für die beschränkten (Speicher-)Möglichkeiten des 16ers sehr gut geraten.



PIN-POINT

In der Nachricht von Kingsoft, mit der wir dieses Spiel erhielten, war Pinpoint ein kurzer Satz gewidmet. Um so größer war unsere Überraschung, nachdem das Spiel im Speicher des 16ers war und gestartet wurde. Denn Pinpoint gefiel uns auf Anhieb durch eine bestechende Grafik und hervorragende Geschwindigkeit.
Wie Sie am Bildschirmfoto erkennen können, handelt es sich bei Pinpoint um eine Art Mischung aus „Marble Madness", welches ja schon längere Zeit für den Commodore Amiga und seit kurzem auch für den C 64 auf dem Markt ist. Ziel ist es, über mehrere Bildschirme hindurch eine Figur per Joystick über Treppen und Absätze zu

ALLES UNTER ZEITLIMIT

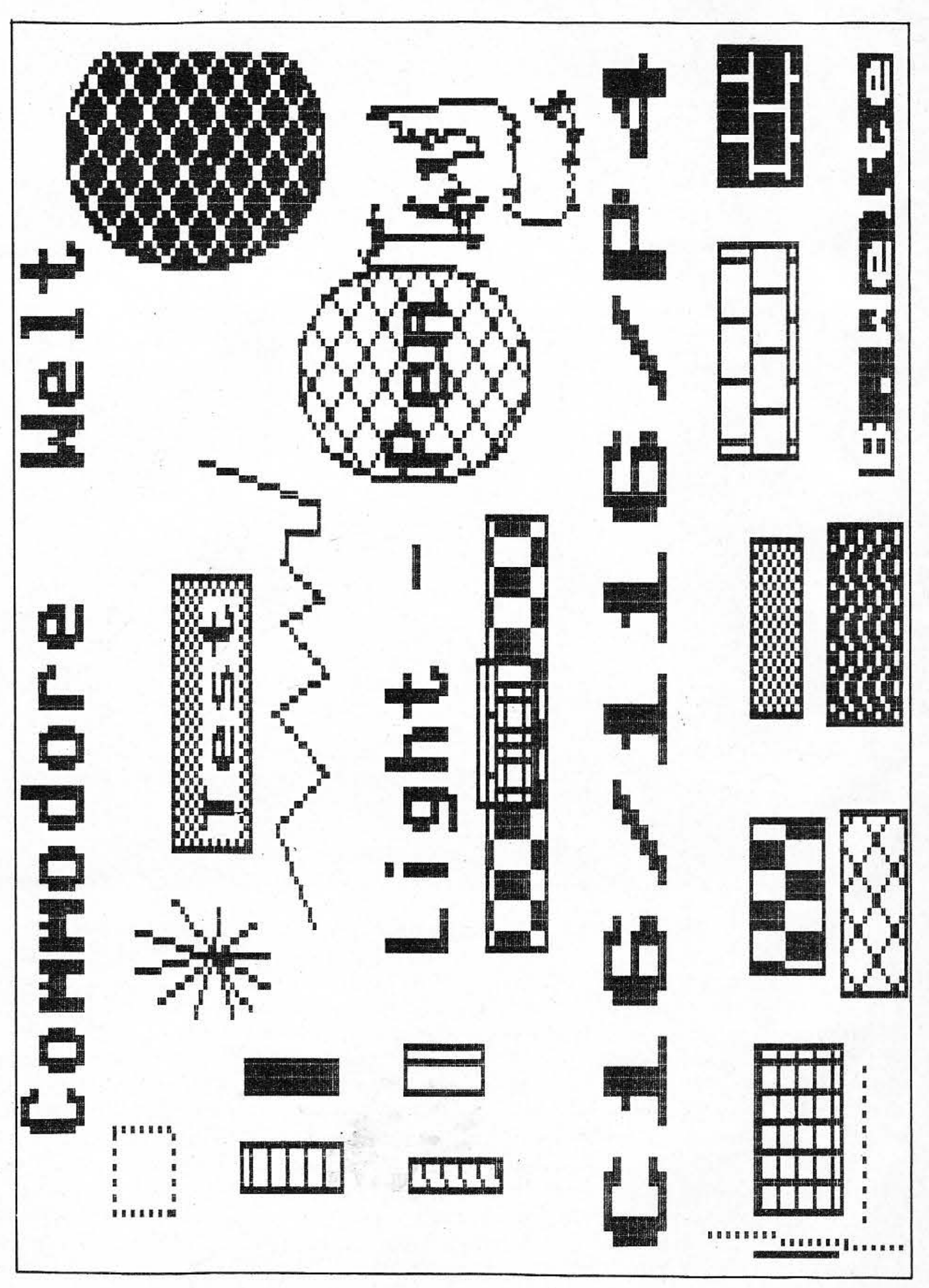
steuern. Das ganze läuft unter Zeitlimit ab, so daß dem Spieler an Aufregung einiges bevorsteht. Die Gemeinheit des Spieles besteht darin, daß man nie vorher weiß, auf welches „Gelände" die Spielfigur als nächstes kommt (außer man hat das ganze schon einige Male durchgespielt). So kann es passieren, daß im nächsten Bild ein steiler Abhang wartet und der Spieler nicht mehr rechtzeitig bremsen kann. Die Figur verschwindet unter leisem Blubbern in der Tiefe und es geht wieder von vorne los. Weiterhin kann man nicht auf dem direkten Weg das Ziel (welches mit einer Fahne gekennzeichnet ist) erreichen. Daher muß man durch die verschiedenen Screens immer wieder hin- und herfahren, um endlich auf den richtigen Pfad zu kommen.
Das Spiel Pinpoint war für uns eigentlich eine der Überraschungen des Monats, denn die Programmierung und Ausnutzung des 16ers kann nur uneingeschränktes Lob verdienen. Hinzu kommt der Preis von 19 Mark, für den Pinpoint von uns die Bewertung „sehr empfehlenswert" bekommt.

PROGRAMM-GESTEUERTER LICHT-GRIFFEL FÜR DEN C16

Fast jede Woche erscheinen neue Artikel auf dem Markt, die auch die Computer berücksichtigen, die nicht so häufig vertreten sind. Neu auf dem Markt für Computerzubehör ist ein Lichtgriffel für den C16/116 und den Plus 4. Obwohl in den verschiedenen Publikationen immer wieder behauptet wird, Lichtgriffel für diese Computer wären nicht möglich, da verschiedene Hardwarevoraussetzungen nicht gegeben seien (fehlender Analogeingang usw.), hat es eine Firma geschafft, einen Lichtgriffel zu entwickeln. Wer sich die Handbücher zu diesen Computern angesehen hat, wird festgestellt haben, daß im TED-Chip hierfür zwei Adressen reserviert sind. Es war also nur eine Frage der Zeit, bis ein solches Gerät auf den Markt kam. Die englische Firma Trojan hat es nun entwickelt, da es speziell auf die oben genannten Computer zugeschnitten ist, hat es auch einen Anschlußstecker, der nur in die entsprechende Buchse am Computer paßt (Joystickport 1). Auf der beiliegenden Kassette befinden sich die Programme, die das Arbeiten mit dem Lichtgriffel ermöglichen. Programme deshalb, weil der C 16 nicht alle Funktionen im Speicher gleichzeitig halten kann. Beim erweiterten C 16 bzw. dem Plus 4 ist dies etwas anders. Hier befindet sich im Gegensatz zum C 16-Programm auch die Druckerroutine, so daß diese nicht nachgeladen wer-

24 MENÜPUNKTE

den muß. Doch nun zum Programm selber. Nach dem Laden und Starten des Programmes erscheint das Hauptmenü, das 24 Menü- und 16 Farbpunkte umfaßt. Bei verschiedenen Menüpunkten erscheint nach dem Anwählen noch ein Untermenü mit weiteren Funktionen. Wie es bei einem Lichtgriffelprogramm üblich sein sollte, werden alle Punkte mit dem Lichtgriffel angewählt. Was kann dieses Programm nun? Eine Menge. Es ermöglicht dem Benutzer, alle Arten von Zeichnungen herzustellen wie andere Zeichenprogramme auch, nur, daß hier die Befehle und Punkte mit dem Lichtgriffel an-bzw. eingegeben werden. Neben den üblichen Möglichkeiten wie Pinselstärke, geometrischen Figuren und Ähnlichem, verschiedenen Strichen, punktiert in allen nur möglichen Richtungen. Und dieses natürlich auch in allen 16 Farben und 7 Helligkeitsstufen. Auch können Teile des Bildes kopiert werden. Die Füllfunktionen ermöglichen ein Füllen der entsprechenden Abschnitte in bis zu 11 Variationen. Einige davon auch negativ. Ein ganz besonderer Leckerbissen verbirgt sich hinter dem Menüpunkt 'TEXT'. Die





TROJAN LIGHTPEN (C)1986

THIN	BOX	DRAW	NEW			
QUIL1	BAND.	BRUSH	ERASE			
QUIL2	RAYS	COPY	BORDR			
QUIL3	LINE	WASH	BRI.2			
THICK	TRIA.	SAVE	PEN1			
XDOT	CIRC.	LOAD	PEN2			
YDOT	FILL	PRINT	PEN3			
DOTLI	TEXT	LOOK	BACK			

PLUS/4 VERSION 1.0

Trojans Lightpen (Vertrieb Ing. Stechmann) bietet sehr viel Komfort

Buchstaben und Zahlen, aber auch die Grafikzeichen der Tastatur, können an allen nur möglichen Stellen des Bildschirmes plaziert werden und zwar in vier verschiedenen Größen. Auch eine negative Darstellung ist möglich. Die Druckerroutine enthält auch eine Besonderheit. Der Ausdruck erfolgt um 90 Grad gedreht, so daß das Papierformat besser ausgenutzt wird. Selbstverständlich lassen sich die erstellten Bilder und Grafiken abspeichern und auch wieder einladen. Natürlich auf Kassette oder Diskette. Alle Möglichkeiten des Programmes aufzuzählen, würde diesen Bericht sprengen, denn es gibt noch viele Möglichkeiten, die es zu entdecken gibt. Alles in allem ein faszinierendes Mal- und Zeichenprogramm, das neue Möglichkeiten auf dem Computer eröffnet. Dazu kommt neben dem umfangreichem Programm auch noch der vergleichsweise günstige Preis von 59,– DM für die Kassetten und 69,– DM für die Diskettenversion. Sollte Ihnen der Preis etwas hoch erscheinen, im Gegensatz zu anderen Lichtgriffeln, sollte man immer noch das hervorragende Programm berücksichtigen, das dem Lichtgriffel beiliegt.

B. Welte

IMPRESSUM

COMMODORE-SPECIAL

erscheint achtmal jährlich (4x C16, 4x 128) in der CA-Verlags GmbH (i.G.)

VERANTWORTLICH FÜR DEN INHALT:
Alfons Mittelmeyer

REDAKTION UND STÄNDIGE MITARBEITER:
Peter Basch, Harald Beiler, Rosemarie Huber, Lothar Miedel, Michael Reppisch, Rudolf Schmid-Fabian, Torsten Seibt, Hermann Wellesen, Bernd Welte

GESCHÄFTSFÜHRER (und verantwortlich für Anzeigen):
Werner E. Seibt

ANSCHRIFT FÜR ALLE VERANTWORTLICHEN:
Postfach 1107,
8044 Unterschleißheim
Tel.: 089/129 80 11
Telex: 5214428 cav-d
Es gilt Preisliste Nr. 7 vom 1.2.1987
Media-Unterlagen bitte anfordern.

VERTRIEB:
Verlagsunion Wiesbaden

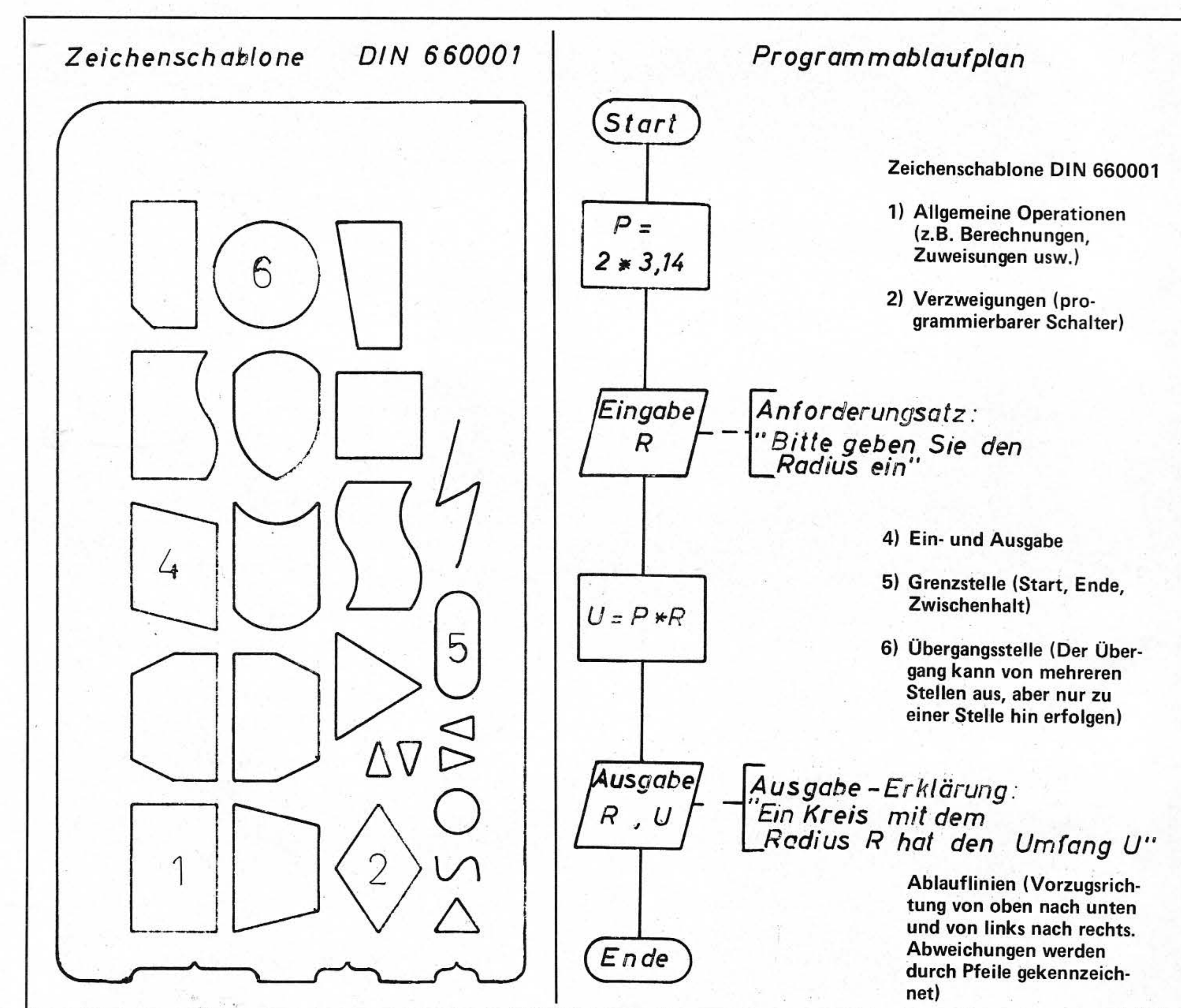
ZUERST MAL EINEN

Geübte Programmierer machen sich vor der Befehlseingabe einen „Ablaufplan". COMMODORE WELT sagt, welche Symbole Sie dabei verwenden können und wie ein solcher Programm-Ablauf-Plan gelesen wird.

rung von Programmen angewendet. Da der Programmablaufplan, in Zukunft PAP genannt, unabhängig von einer Programmiersprache ist, kann auch ein Laie sehr schnell den Programmablauf, bzw. das Programmschema erkennen. Auch kann ein PAP rela-

zum Lieferumfang. Damit ist gewährleistet, daß auch noch nach Jahren eine Programmänderung bzw. Anpassung möglich ist. Für jemanden, der noch nichts mit PAP's zu tun hatte, ist es meist sehr schwer, seine Ideen und Programme in Form von PAP's darzustellen.

ne, die das Zeichnen der Symbole erleichter. Diese Schablone erhält man in Zeichenfachgeschäften und EDV-Abteilungen der Warenhäuser und Fachgeschäfte.
In Bild 1 ist so eine Schablone abgebildet.
Diese Symbole sind nach DIN 660001 genormt



Zeichenschablone DIN 660001

1) Allgemeine Operationen (z.B. Berechnungen, Zuweisungen usw.)

2) Verzweigungen (programmierbarer Schalter)

4) Ein- und Ausgabe

5) Grenzstelle (Start, Ende, Zwischenhalt)

6) Übergangsstelle (Der Übergang kann von mehreren Stellen aus, aber nur zu einer Stelle hin erfolgen)

Ablauflinien (Vorzugsrichtung von oben nach unten und von links nach rechts. Abweichungen werden durch Pfeile gekennzeichnet)

Der Programmablaufplan als Hilfsmittel zur Programmerstellung und Dokumentation wird von jeher von Informatikern und Programmierern zur Erklärung und Erläute-

tiv einfach in die jeweilige Programmiersprache umgesetzt werden. Bei vielen kommerziellen Programmen gehört außer anderen Dokumentationen auch ein PAP

Deshalb nun einige Tips, die den Umgang mit PAP's, erleichtern. Um einen PAP zu erstellen, benötigt man außer den Kenntnissen der Symbole auch eine Schablo-

und bedeuten im einzelnen:
Erklärung der gängigsten Symbole.
1) Allgemeine Operationen (z.B. Berechnungen, Zuweisungen usw.)

PLAN!

2) Verzweigungen (programmierbarer Schalter)
3) Unterprogramm
4) Ein- und Ausgabe
5) Grenzstelle (Start, Ende, Zwischenhalt)
6) Übergangsstelle (der Übergang kann von mehreren Stellen aus, aber nur zu einer Stelle hin erfolgen)
7) Ablauflinien (Vorzugsrichtung von oben nach unten und von links nach rechts. Abweichungen werden durch Pfeile gekennzeichnet)
8) Bemerkung (zur erweiterten Erklärung).

Nach der vielen Theorie schreiten wir nun zur Tat. Um einen PAP zu erstellen, muß zuerst ein Konzept über die Tätigkeit des Programmes vorliegen.

Beispiel Nr. 1

Ein Computerfreak möchte sich einen neuen Computer kaufen. Dieses ist aber von zwei Faktoren abhängig. 1. von einer Lohnerhöhung von mindestens 5% und einer Preissenkung von mindestens 100,– DM. Sollte eine dieser Faktoren nicht zutreffen, soll das Programm vom Kauf abraten.

Nun zur Erklärung des PAP. Jeder Lösungsweg hat zwei Grenzstellen, die durch ein Oval gekennzeichnet sind. Nach dem Start kommen wir zu einem Rechteck, das für die allgemeine Operation steht. Hier werden die Variablen P und L festgelegt. Da man das Rechteck nicht unbegrenzt groß zeichnen kann, greift man zu einem Hilfsmittel, nämlich der eckigen Klammer.

Ein- und Ausgaben werden mit dem Parallelogramm dargestellt.

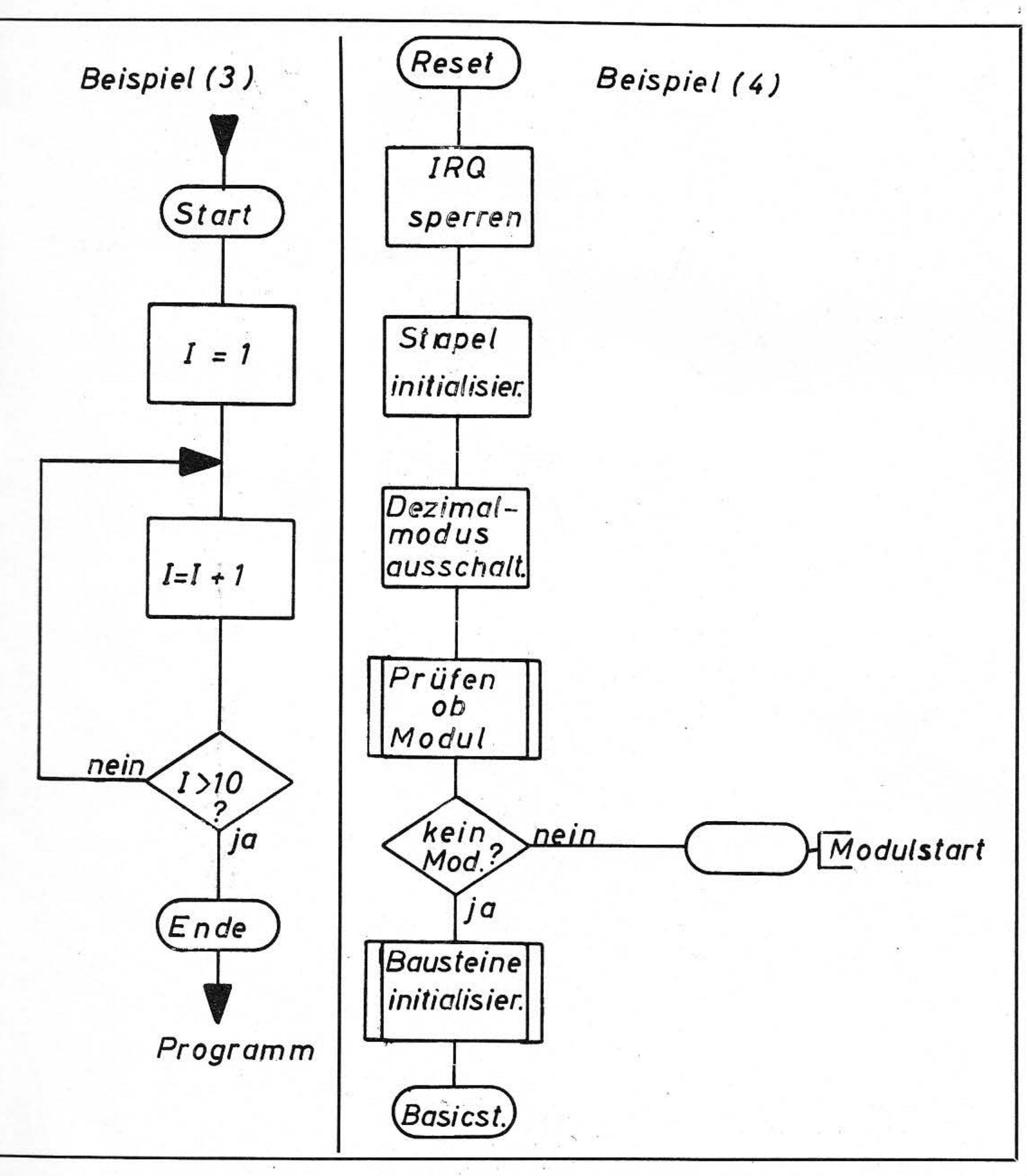
Für Verzweigungen und logische Entscheidungen findet die Raute Verwendung. Während die anderen Symbole nur einen Ein- und Ausgang haben, hat die Raute zwei alternative Ausgänge, mit den logischen Entscheidungen JA oder NEIN. Verbunden werden die Symbole mit den Ablauflinien, die mit Pfeilen gekennzeichnet werden (Laufrichtung), aber nicht müssen. Nur wenn die Linien von rechts nach links und von unten nach oben verlaufen, müssen Pfeile angebracht werden. Erstreckt sich ein PAP über mehrere Seiten, ermöglichen die Konektoren, das sind mit Zahlen versehene Kreise, ein leichtes Wiederfinden der weiterführenden Linie.

Beispiel Nr. 2

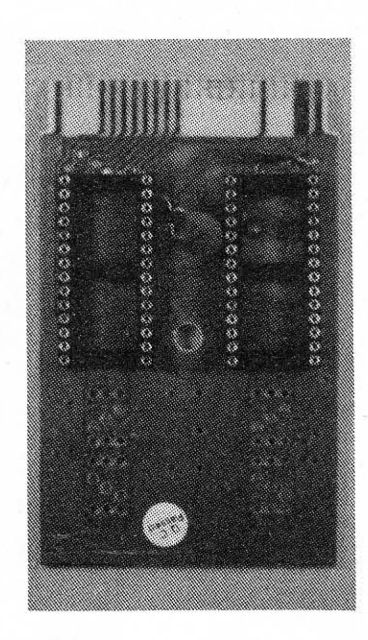
Hier soll in einem Programm der Umfang eines Kreises, bei gegebenem Radius errechnet werden. Die Gegenüberstellung mit dem entsprechenden Basic-Programm zeigt, wie einfach eine Umsetzung eines PAP in eine Programmiersprache ist.

Beispiel Nr. 3 zeigt eine FOR-NEXT-Schleife, wie sie immer wieder als Zähl- oder Warteschleife vorkommt als PAP und Basic-Listing.

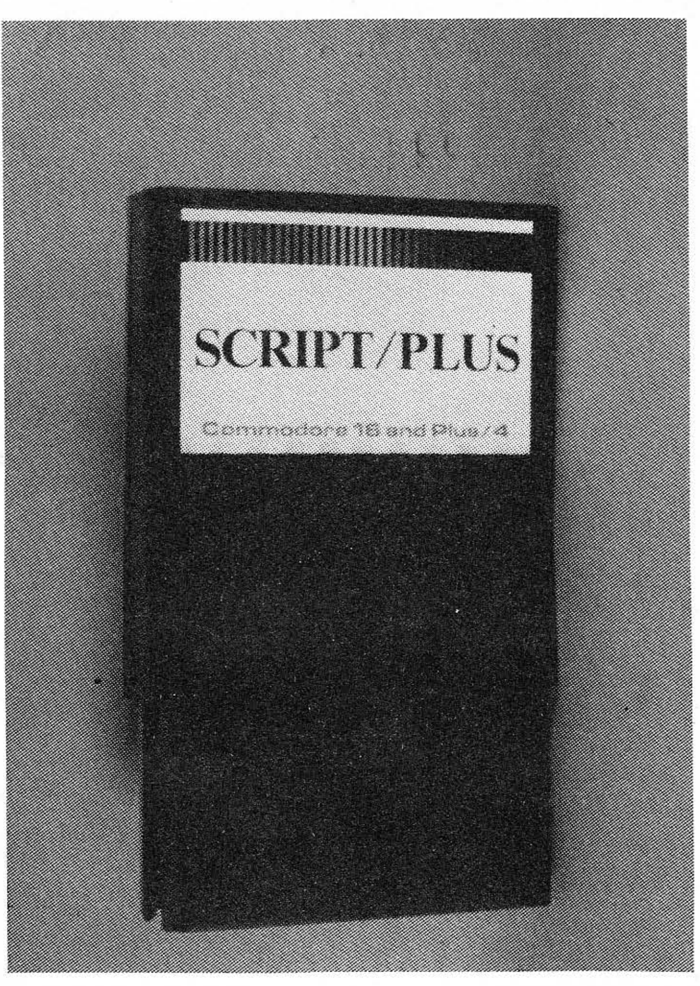
Als letztes Beispiel noch ein PAP, in dem ein Reset des Computers dargestellt wird. Dieser Ablauf findet auch bei jedem Einschalten des Computers statt. Fürs erste soll es genug sein. Sollten Sie irgendwelche Fragen haben, schreiben Sie an die Redaktion. Nun kann ich Ihnen nur noch Hals- und Diskettenbruch wünschen.

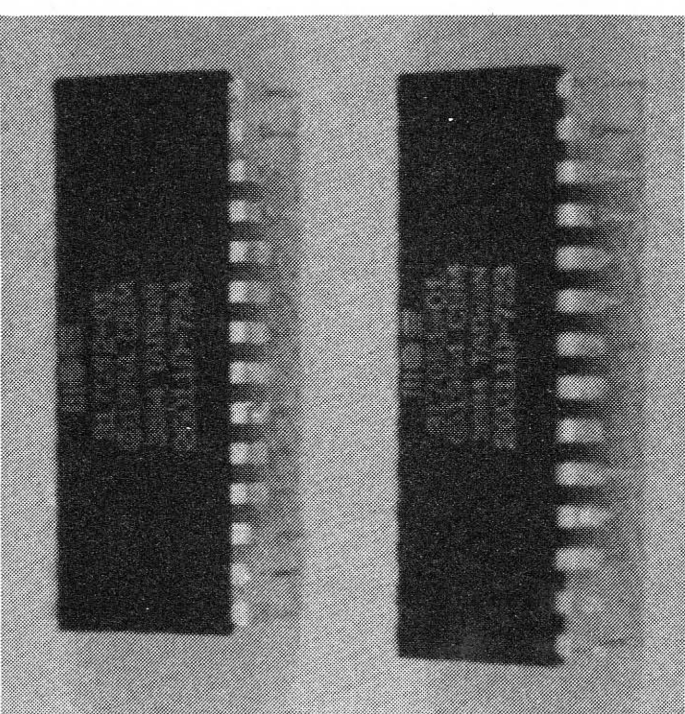


SCRIPT PLUS ALS ROM IM PLUS 4!



Script/Plus im Plux/4? Was ist das? Im Plus/4 ist ein Textprogramm eingebaut und Script/ Plus liegt nur als Modul vor. Wie kommt nun das Modul in den Computer? Vorneweg etwas zur Einführung. Das Textprogramm Script/Plus belegt als Modul den Expansionsport und blockiert somit diesen für andere Anwendungen. Auch die Funktionstaste ‚F2' ist mit dem Startbefehl belegt. Das führt immer wieder zu Mißverständnissen, da normalerweise mit ihr der oft gebrauchte Befehl 'DLOAD" ' verbunden ist. Also suchte ich nach einer anderen Lösung. Es müßte doch möglich sein, dieses Programm, das sich in zwei ROM auf der Modulplatine befindet, gegen die zwei ROM mit der eingebauten Software auszutauschen.



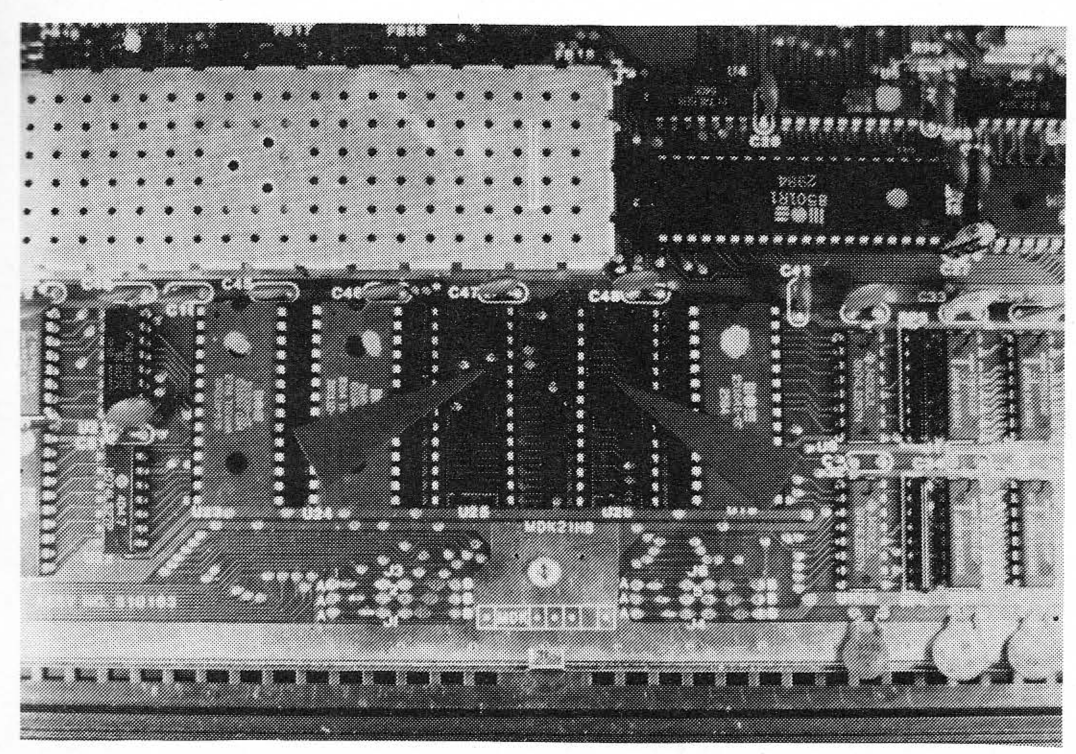


Die Module von Script Plus müssen sehr vorsichtig ausgelötet werden (umgeknickte Beinchen!)...

Nach dem Öffnen der beiden Gehäuse, Computer und Modul, sowie gründlichem Studium der Aufschriften mit anschließendem Kennzeichnen der Bauteile auf richtige Position und Lage konnte es losgehen.
Im Gegensatz zum Computer, wo die ROM gesockelt sind, sind diese auf der Modulplatine eingelötet und zwar in durchkontaktierten Leiterbahnen, womit sich das Auslöten relativ schwierig gestaltet. Um die Sache noch zu erschweren, sind auch noch einige Pins auf der Lötseite umgebogen.
Je sauberer und gewissenhafter diese Arbeit

SEHR SORGFÄLTIG ARBEITEN!

ausgeführt wird, um so größer ist die Gewißheit, daß sie nicht elektrisch oder mechanisch beschädigt werden, denn sie sollen ja noch im Plus/4 ihren Dienst verrichten. Nachdem Sie nun die ROM sauber ausgelötet haben, können Sie in die Modulplatine Fassungen einlöten, in die dann die Original ROM eingesetzt werden können. So kann man diese weiter benutzen. Der Start erfolgt dann mit der 'F2'-Taste. Diese Modulkarte läuft auch im C 16/116 ohne Störungen. Dieses nur nebenbei. Doch nun zum Ausbau der Chips aus dem Plus/4. Die eingebaute Software befindet sich in den Chips U25 und U26. Auch hier hilft eine Kennzeichnung weiter, damit die Chips nicht vertauscht werden, denn sie sind auch wie beim Script/Plus in LOW und HI eingeteilt. Dieses ist an den Zahlen auf den Chips zu erkennen. Es gibt dort eine höhere


...und passen dann exakt anstelle der Plus 4-Text-Roms

und eine niedrige Zahl, die um eine Stelle auseinander liegen. Genau ansehen. Auch ist auf die richtige Polung zu achten (Kerbe am Chipgehäuse). Nachdem nun die beiden Chips aus den Fassungen gehebelt sind, können die beiden Script/Plus-Chips eingesetzt werden. Auch hier trifft oben gesagtes zu (hohe Zahl auf hohe Zahl, niedrige auf niedrige). Sind die Chips nun ordnungsgemäß in die Fassungen eingesetzt (auf umgebogene Beinchen achten), kann eine Funktionsprüfung vorgenommen werden. Auf dem Bildschirm sollte nun, zusätzlich zur normalen Einschaltmeldung, die Meldung 'Script/Plus on Funktion Key F1' in der Zeile 3 erscheinen.

Ist dieses der Fall, haben Sie es geschafft. Wenn nicht, überprüfen Sie, ob auch wirklich alle Beinchen richtig in den Fassungen stecken. Überprüfen Sie auch die richtige Reihenfolge der Chips. In Sockel U25 kommt Chip Nr. 1 von der Modulplatine und in Sockel U26 Nr. 2. Diese Bezeichnung befindet sich auf der Modulplatine. Normalerweise ist der Umbau für jemanden, der etwas Elektronikerfahrung besitzt und weiß, an welchem Ende der Lötkolben heiß wird, nicht schwierig und sollte auch zum Erfolg führen. Wenn Sie nun mit Ihrem Plus/4 arbeiten, werden Sie feststellen, daß dieses etwas einfacher geworden ist. Auch die 'F2'-Taste ist nicht mehr blockiert, außerdem haben Sie nun einen freien Expansionsport, der nun für andere Zwecke zur Verfügung steht (Centronicschnittstelle usw.). Bei Bedarf kann man ja noch das Modul mit der neuen (alten) Software an ihm betreiben.

B. W.

LAUTSPRECHER AM COMPUTER

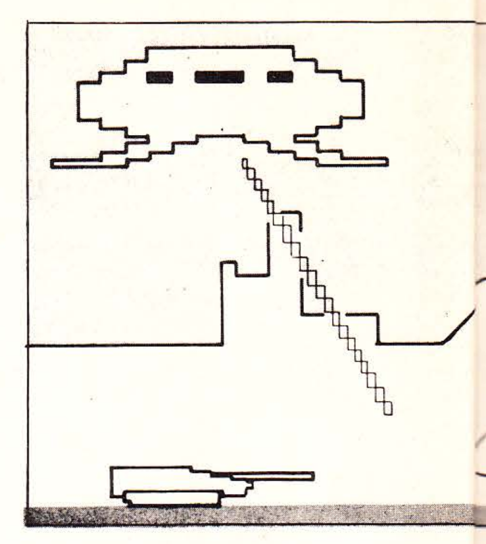
Auch in der heutigen Zeit haben viele Computerbesitzer noch Probleme hinsichtlich der Tonausgabe mit ihrem Monitor. Daß die meisten Computer einen Soundchip besitzen, hat sich wohl noch nicht zu allen Monitorherstellern herumgesprochen, denn wie kann es sonst möglich sein, Monitore, selbst der höheren Preisklasse, ohne Tonwiedergabe auf den Markt zu bringen? Dieses rührt wahrscheinlich daher, weil die Monitore für PCs oder ähnliches gedacht sind, die mit einem eingebauten Lautsprecher von Haus aus bestückt sind. Nicht so die meisten Homecomputer. Obwohl es auch hier die eine oder andere Ausnahme gibt. Selbst der so hochgepriesene ,Amiga', der seine Töne sogar stereo der Umwelt mitteilen kann, ist mit einem Monitor ausgerüstet, der nur eine einkanalige Tonausgabe zuläßt. Für Stereoausgabe empfiehlt Commodore die Benutzung einer Stereoanlage. Doch auch hier gibt es Probleme, denn meistens befindet sich die Anlage an einer ganz anderen Stelle der Wohnung. Was ist also zu tun? Den Monitor mit einem (zweiten) Tonteil ausstatten? Oder gibt es noch eine andere Möglichkeit? Ja. Auf dem Markt für Audiogeräte gibt es von verschiedenen Herstellern sogenannte Aktivboxen, das sind Lautsprecher mit eingebautem Verstärker. Diese Aktivboxen eignen sich, durch ihren Verstärker, hervorragend für den Anschluß an den Computer. Mit einem einfachen Kabel lassen sie sich an jeden Computer anschließen. Ein Nachteil soll aber nicht verschwiegen werden. Einige dieser Boxen haben, genau wie die Computer auch, keinen Lautstärkeregler. Aber das soll der Freude keinen Abbruch tun. In der Zeichnung finden Sie eine einfache Möglichkeit, einen Lautstärkeregler in die Leitung zwischen Computer und Lautsprecherbox einzubauen. Wer geschickt ist, kann den Regler auch in die Box einbauen.
Auch der Einbau in den Computer ist möglich. An Material benötigt man nur ein Stück einadriges abgeschirmtes Kabel, einen normalen Diodenstecker (für Commodore), ein Poti von ca. 20–25 kOhm mono oder stereo, z.B. für den Amiga und einen oder zwei Klinkenstecker 3,5 mm mono. Die Anschlüsse entnehmen Sie bitte dem Schaltplan. Um den Regler in die Box einzubauen, sollten Sie etwas Elektronikerfahrung besitzen, da hierzu die Box zerlegt werden muß. Auch müssen eine oder auch mehrere Leiterbahnen oder Leitungen unterbrochen und die drei Drähte zum Poti angelötet werden. Ist die Box mit einem Regler ausgestattet, entfällt dies und Sie brauchen sich nur ein Kabel mit entsprechenden Steckern für Ihren Computer anzufertigen. Sollten Sie alles erfolgreich hinter sich gebracht haben, werden Sie den Soundchip in ganz neuem Licht, Pardon Ton, erleben.

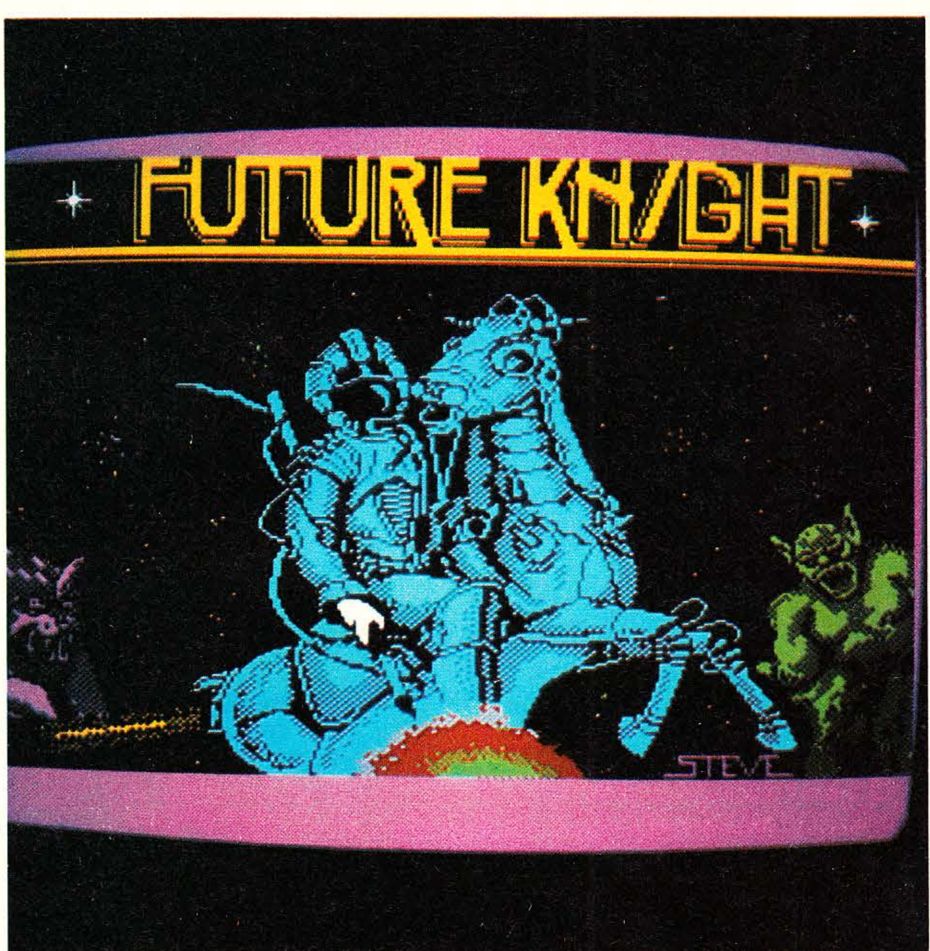
B.W.

NEUE SPIELE FÜR C16/116/PLUS 4

ZAUDERE NICHT GROSSER HELD!

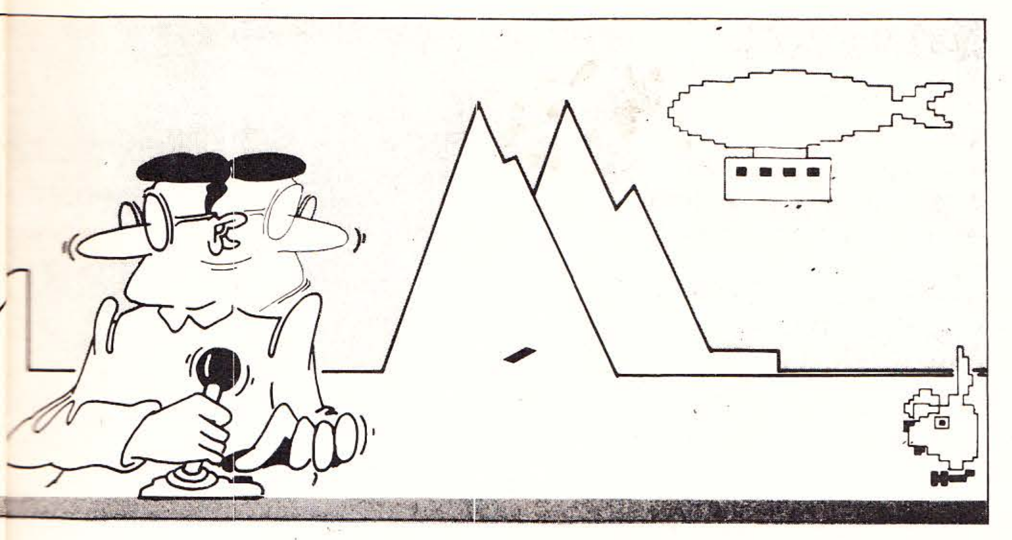
Als heldenhafter Ritter, mit Namen Randolph, müssen Sie die Prinzessin Amelia befreien. Diese ist vom bösen Zauberer Spegbot geraubt worden, sitzt nun in seiner Festung und schmachtet dahin.
Mit einem Mehrzweckkampfanzug und einem Plasmagewehr bewaffnet, eilt Randolph zu der nächstgelegenen Transporterstation und findet sich einige Sekunden später in dem notgelandeten Raumschiff seiner Geliebten wieder.
Hier beginnt die Mission für den Spieler, der jetzt in die Rolle von Randolph schlüpft.
Der Spieler muß sich, um seine Geliebte zu befreien, den Kreaturen Spegbots, die es in einer großen Vielzahl gibt, entgegenstellen.
Um durch das Raumschiff auf den Planeten zu gelangen, müssen zwanzig Ebenen durchspielt werden, was gar nicht einfach ist, denn der Spieler hat nur eine begrenzte Lebensenergie zur Verfügung. Diese schwindet bei Berührung mit einer Kreatur und mit dem Verrinnen der Zeit.
Zur Lösung der Aufgabe stehen bestimmte Gegenstände zur Verfügung, die aufgenommen und mit der „U"-Taste eingesetzt werden können.
Es gibt zwei Arten von Gegenständen, die dem Spieler das Durchqueren des Raumschiffes erleichtern, indem die Monster verwirrt oder zerstört werden. Die anderen ermöglichen den Zugang zu einem anderen Teil des Schiffes.
Wenn die zwanzig Ebenen erfolgreich durchquert wurden, steht der Spieler den Berzerka-Droiden gegenüber, diese versperren den Durchgang zum Planeten. Hat der Spieler auch sie überwunden, muß er zum Schloß eilen, sich immer noch gegen Kreaturen wehrend, um endlich zu seiner Geliebten zu kommen, die von dem übermächtigen Henkerdroiden bewacht wird. Jetzt beginnt ein Kampf auf Leben und Tod, falls einen dieser nicht schon vorher ereilt hat.
Dieses Spiel kann mit Joystick oder Tastatur (Cursor-Tasten) gespielt werden. Während des Spieles hat der Spieler die Möglichkeit, die Waffen zu wechseln.
Die oberen zwei Drittel des Bildschirmes werden für das Spielfeld benötigt, das untere Drittel für die Anzeigen. Rechts ist die verbleibende Energie zu sehen, in der Mitte die Gegenstände, die Randolph mit sich herumträgt, und links die verbleibenden Leben.





Future Knight: Armer Ritter

Noch einmal zurück zu den Gegenständen. Einige können nur in bestimmten Räumen benutzt werden, andere kann Randolph nur allein mit sich herumtragen. Es sollte also immer darauf geachtet werden, welcher Gegenstand von Nutzen sein kann und welcher nicht.
Wenn Randolph länger nicht bewegt wurde, so beginnt er auf dem Bildschirm mit den Füßen zu stampfen, wird er immer noch nicht bewegt, so fängt er zusätzlich an, mit den Armen zu winken. Grafisch ist das Spiel auf jeden Fall sehr gut aufbereitet, die Figuren sind groß, aber nicht klobig und sehr genau ausgearbeitet.
Future Knight, so der offizielle Name des Spieles, bietet dem Spieler neben einer sehr schön inszenierten Spielidee auch eine tolle Grafik.





Tiger: Besser als Kung Fu

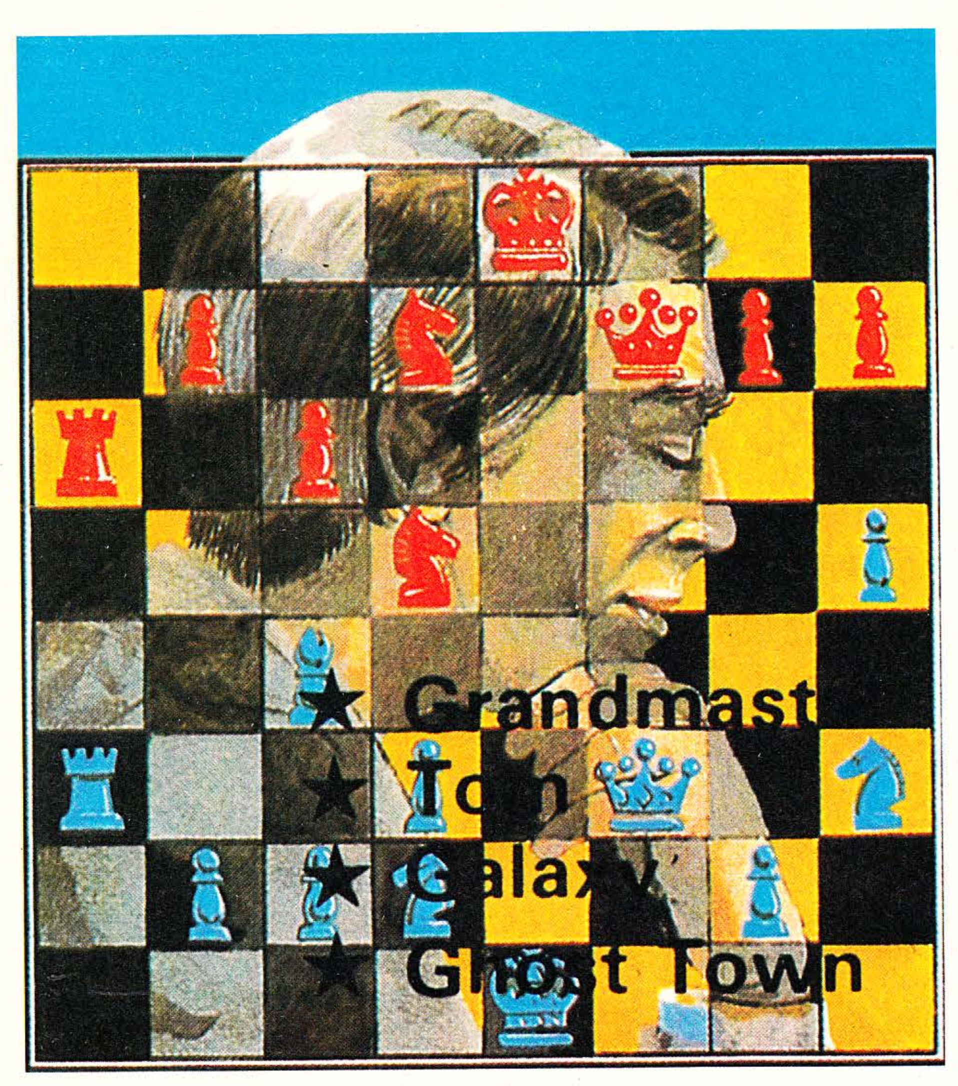
DER WEG DES TIGERS

Das gleichnamige Spiel von Gremlin Graphics ist jetzt auch für C16/116 auf dem Markt. Es erscheinen auf dem Bildschirm vier Auswahlkriterien. Zur Übung kann jede Kampfart einzeln oder die ganze Reise, angefangen von Unarmed Combat bis Samurai Sword Fighting, ausgewählt werden.
Wurde ein Punkt ausgewählt, so erscheint die Mitteilung, die Programmkassette mit der gleichnamigen Aufschrift in den Recorder zu legen und 'Play' zu drücken.
Ort der Handlung ist die Insel der stillen Tempel, die sich in Orb, einer Zauberwelt in einer Meeresöde, befindet.
Als Waise wurden Sie von einer Amme, die von einem Fluch verfolgt wurde und schon im Sterben lag, auf den Stufen des Felsentempels abgelegt. An jenem Ort wurden Sie von Naijishi, dem Großmeister der Dämmerung und obersten Mönch von Kwon – Gott des unbewaffneten Kampfes – gefunden und aufgezogen.
Er lehrte und schulte Sie in der stillen Kunst von Kwon. Auf Ihre Fragen wegen eines Muttermales auf Ihrem Oberschenkel ging kein Mönch ein.
Als die Zeit reif war, die Große Prüfung zu bestehen, schickte Sie Naijishi auf den Weg um die Insel, um gegen die besten Krieger anzutreten, die Ihnen von Naijishi gesandt werden.
An dieser Stelle beginnt es ernst für den Spieler zu werden. Um sich verteidigen zu können, können die Cursor-Tasten oder der Joystick benutzt werden.
In der untersten Zeile des Bildschirmes sind die innere Kraft und Ausdauer von Ihrem Gegner (links) und von Ihnen selbst (rechts) zu erkennen. Wenn ein Gegner überwältigt wurde, so erhöht sich Ihre innere Kraft, bei jedem Treffer geht etwas innere Kraft und Ausdauer verloren. Wenn eine Kampfart erfolgreich überwunden wurde, so geht es zu der nächsten. Am Schluß müssen Sie gegen Ihren eigenen Lehrer kämpfen, wird dieser von Ihnen überwunden, so können Sie sich Ninja – Mächtigster der Welt – nennen.
Aber bitte: Dies ist ein Spiel und nicht die Wirklichkeit.

Tip für das Spiel:
Beim Stockkampf sollten wenig Hiebe, sondern mehr Stöße ausgeteilt werden, weil diese in einer rascheren Folge abgesetzt werden können.

FAZIT:

Ein tolles Spiel, das sich aus dem Wirrwarr der Schieß- und Strategiespiele hervortut. Alleine schon die Figuren sind beneidenswert gut dargestellt. Wenn einer, egal ob Gegner oder seine eigene Figur, schwer getroffen wurde, breitet die Figur die Arme aus und fliegt nach hinten.
Falls ein Vergleich mit Konamis Kung Fu gestattet sein sollte, so muß dieses Spiel hinter „The way of the Tiger" zurückstehen.





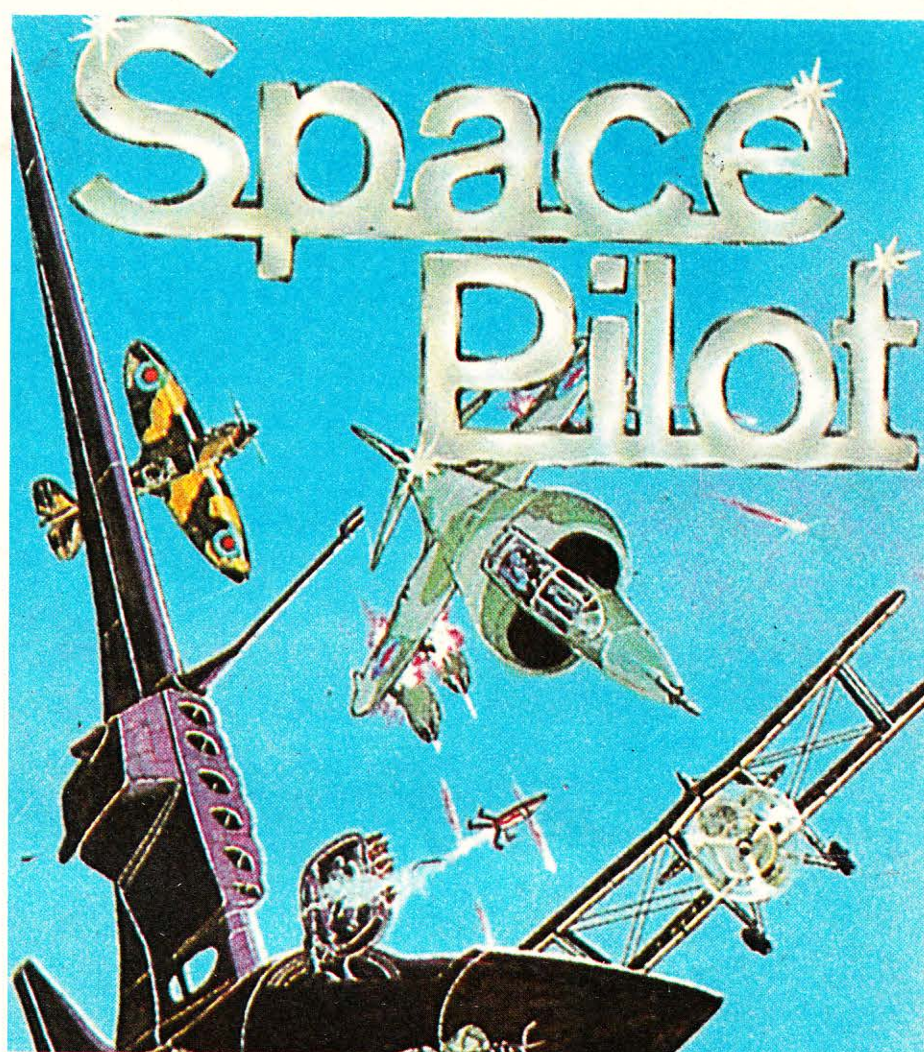
KINGSOFT: WER SAGT DENN, DASS NUR DIE ENGLÄNDER GUTE PROGRAMME HABEN?

Daß in England hervorragende Programmautoren beheimatet sind, ist inzwischen wohl hinlänglich bekannt. Das deutsche Softwarehaus Kingsoft allerdings beweist, daß sich auch die inländischen Programmierer nicht zu verstecken brauchen. Kingsoft bietet eine ganze Reihe von Spielen auf den 3.5er-Modellen an. Einige hiervon im Test.

SPACE PILOT

Ein weiteres Abenteuer im Weltraum. Von oben wird das fliegende Schiff dargestellt, dessen Aufgabe es ist, verstreut umherstreifende Gegner aufzuspüren und zu vernichten. Vom Bildschirmaufbau und der Grafik her absolut hervorragend, muß dieses Spiel sich jedoch auch Kritik gefallen lassen: Die Steuerung ist äußerst gewöhnungsbedürftig und in puncto Geschwin-







digkeit stellt Galaxy mit Sicherheit keine Rekorde auf.
Trotzdem ein interessantes Spiel.
Empfehlenswert

GRANDMASTER

Schachsimulation. Ein hervorragendes Schachprogramm für die C16/C116/Plus 4-Reihe hat Kingsoft auch zu bieten.
Natürlich sind alle Möglichkeiten anderer Schachsimulationen ebenfalls enthalten (z.B. Tip oder Erstellen einer eigenen Partie). Für den Schachliebhaber und C 16–Besitzer ein Muß!
Sehr empfehlenswert

BONGO

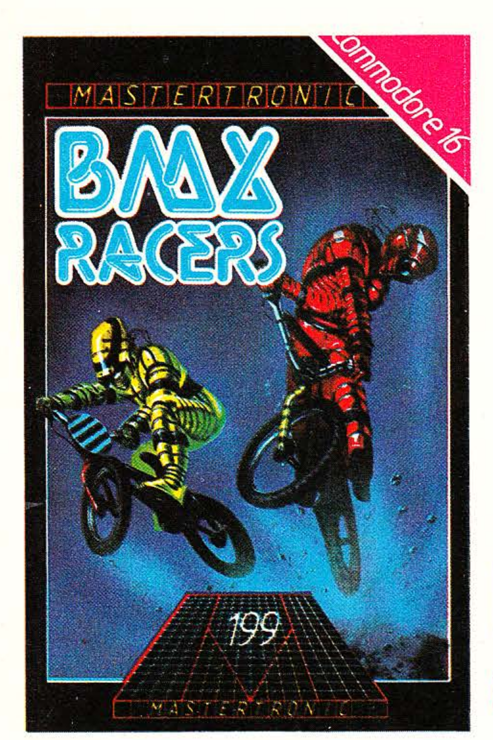
Auf einem Klettergerüst mit Rutschen, Leitern und Aufzügen muß die Spielfigur versuchen, herumfliegende Buchstaben aufzusammeln, um den Namen "Bongo" zu buchstabieren. Hierbei kommt dem Helden aber ein langnasiger Schurke ins Gehege, welcher nichts Besseres zu tun hat, als diesen vom Gerüst zu werfen.
Dieses Spiel stellt in jeder Hinsicht eine Bereicherung der eigenen Sammlung dar. Angefangen vom absolut gelungenen Sound, als Anfangsmelodie wird das bekannte "Axel F."-Stück gespielt, über die hervorragende Grafik bis hin zur Spielidee und Durchführung verdient Bongo uneingeschränktes Lob.
Sehr empfehlenswert

GHOST TOWN

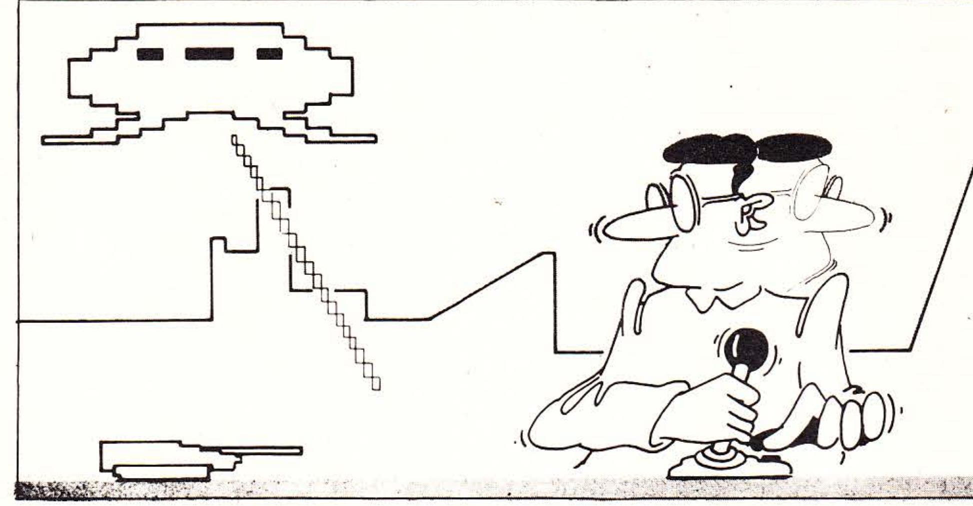
In einem unterirdischen Labyrinth muß durch viele Räume hindurch versucht werden, ein geheimes Lösungswort zu finden, um den bösen Zauberer zu vernichten. Die Grafik ist recht gelungen, jedoch kommt es durch die großräumige Steuerung öfters zu Kollisionen mit Dornenbüschen und dergleichen, was zum Spielende führt. Daher ist bei Ghost Town einige Übung erforderlich, um dem Ziel wenigstens auf Reichweite nahe zu kommen. Der Spaß bleibt mit Sicherheit über längere Zeit ungetrübt.
Empfehlenswert

GALAXY

Die inzwischen wohl ausreichend ausgelutschte Idee vom einsamen Raumschiff, welches sich unbarmherzigen Angreifern stellen muß, ist hier zu einem doch recht gelungenen Spiel verarbeitet worden. In bekannter Weise treibt das eigene Schiff am unteren Bildschirmrand hin und her und versucht, die Angreifer, welche in immer neuen Formationen und Variationen angreifen, zu vernichten. Da tauchen ebenso furchterregende wie unbarmherzige Elmorcs auf, auch mit Forculs (hoffentlich weiß wenigstens der Autor, was die ganzen Namen bedeuten) darf sich der einsame Streiter beschäftigen. Ein recht gelungenes Spiel, welches trotz der bekannten Idee durch Abwechslungsreichtum besticht. Uns jedenfalls ist es nicht gelungen, irgendeine Angreiferformation wiederholt zu sichten.
Empfehlenswert



BMX Racers



NEUE SPIELE GESICHTET!

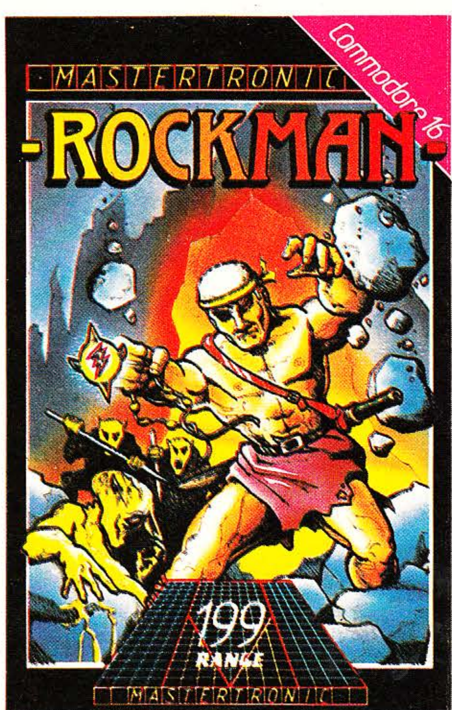
Mit wachsender Verbreitung der 3.5er-Rechner von Commodore wächst auch das Angebot an Spielesoftware. Zwar ist dieses immer

THE EXPLOITS OF FINGERS MALONE

Instruktionen: Englisch.
Die Spielfigur muß in einem Klettergerüst mit Aufzügen verschiedene Schlüssel einsammeln, um mit diesen einen Safe knacken zu können. Hierbei sind allerdings zwei Minimonster sehr hinderlich, welche den Safeknacker bei Berührung verspeisen. Ist ein Safe geöffnet, gelangt die Figur in die nächste Runde von 15. Das Spiel läuft sehr schnell ab, so daß man anfangs kaum über die erste Spielstufe hinauskommt. Die Grafik muß als mittelmäßig bezeichnet werden, allerdings bleibt der Spielspaß davon unberührt.
Ein weiteres Spiel auf dem "Baugerüst".
Bedingt empfehlenswert

ROCKMAN + THE RETURN OF ROCKMAN

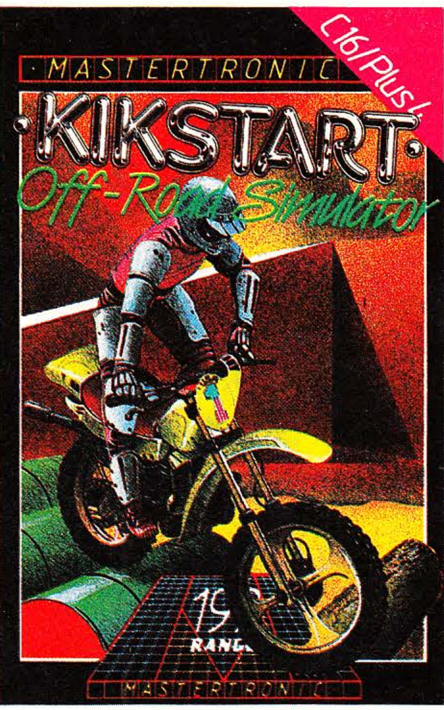
Instruktionen: Deutsch.
Die Figur muß, ähnlich dem bekannten Automatenspiel, in unterirdischen Stollen Diamanten aufsammeln. Gräbt er dabei zu nahe an den verstreuten Felsbrocken, fallen diese hinunter und bei zu langsamer Reaktion ihm auf den Kopf.
Dieses Spiel kommt dem bekannten Original schon recht nahe und ist auch über längere Dauer interessant. Die Fortsetzung des Spieles, "The Return of Rockman", ähnelt der ersten Version sehr stark, lediglich der Bildschirmaufbau variiert etwas. Mit der Grafik hätte man sich im Hause Mastertronic zwar etwas mehr Mühe geben können, das beeinträchtigt den Spaß jedoch kaum.
Empfehlenswert



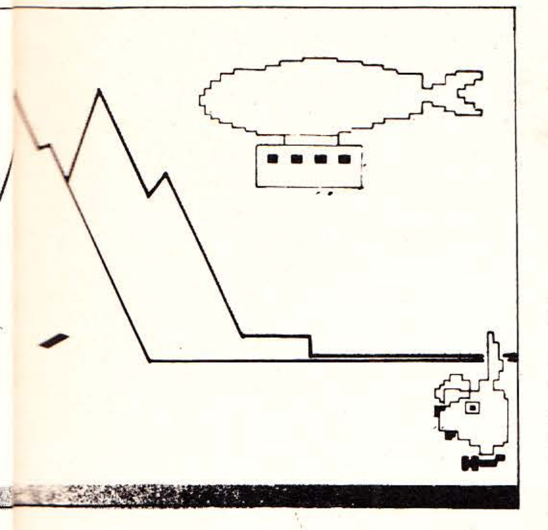
Rockman

VEGAS JACKPOT

Instruktionen: Deutsch.
Dies ist eine absolut gelungene Nachempfindung der bekannten einarmigen Banditen aus Las Vegas. Auf vier verschiedenen Scheiben rotieren Früchte mit unterschiedlicher Bedeutung und damit auch



Kikstart



noch meilenweit vom legendären C 64 entfernt, es ist jedoch offensichtlich, daß auch für C 16/116/ Plus 4-Eigner rosige Spielzeiten angebrochen sind. Wir nahmen einige Spiele der beiden großen Anbieter auf dem deutschen Markt, Kingsoft und Mastertronic, unter die Lupe.

unterschiedlicher Gewinnmöglichkeit. Zuerst zählen natürlich mehrere gleiche Früchte in Reihenfolge. Hierdurch gelangt man auf die "Risikoleiste", auf welcher ständig der mögliche Gewinn hin- und herblinkt. Geht man das Risiko ein, kann der Gewinn bis zu verzehnfacht werden, allerdings ist es natürlich auch möglich, daß der Spieler leer ausgeht. Wer das Risiko scheut, kann auch den kleinen, aber sicheren Betrag übernehmen. Stehen neben den Früchten Zahlen und beträgt die Gesamtsumme mehr als sechs, gelangt man in die obere Risikoleiste, in der man sich für die Gewinnausspielung ähnlich wie vorher qualifizieren kann. Erreicht man hierbei durch Übernehmen oder eben durch Risikobereitschaft eine Nummer, darf in der unteren Risikoleiste der Gewinn erhöht oder übernommen werden. Je nach der oben erreichten Nummer gelangt man unten in eine entsprechend vorgerückte Position auf der Gewinnskala, in der dann wie anfangs beschrieben gewonnen werden kann. Der Anfangsbesitz liegt bei 100 Pfund, erreicht der Spieler 250, so hat er den Jackpot gewonnen. Mit der überaus gelungenen Grafik und der wirklichkeitsgetreuen Funktionsweise stellt dieses Spiel einen echten Ersatz für diejenigen dar, die schon immer gerne einen "Einarmigen" im Wohnzimmer haben wollten.

Sehr empfehlenswert

PS: Leider ist es uns bisher noch nicht gelungen, den Computer zur Auszahlung von Bargeld zu bewegen.

SPECTIPEDE

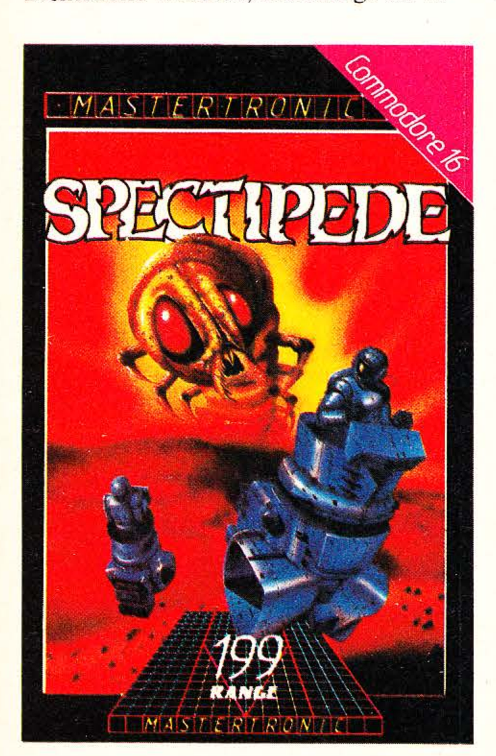
Instruktionen: Deutsch.
Dieses Spiel (der Name verrät es bereits) ist dem bekannten Klassiker Centipede nachempfunden. Es müssen durch verschiedene Schwierigkeitsstufen hindurch alle möglichen Krabbeltierchen, wie Spinnen usw., beseitigt werden. Erschwert wird dies durch Hindernisse, welche abgeschossen werden müssen und durch den "Plasmabolzen" ständig wieder aufgebaut werden. Dieser ist übrigens resistent gegen sämtliche Schüsse, außerdem reflektiert er diese gemeinerweise. Durch die hervorragende Grafik und sehr hohe Spielgeschwindigkeit ist dies ein gelungener "Nachbau" von Centipede auf dem C 16.
Sehr empfehlenswert



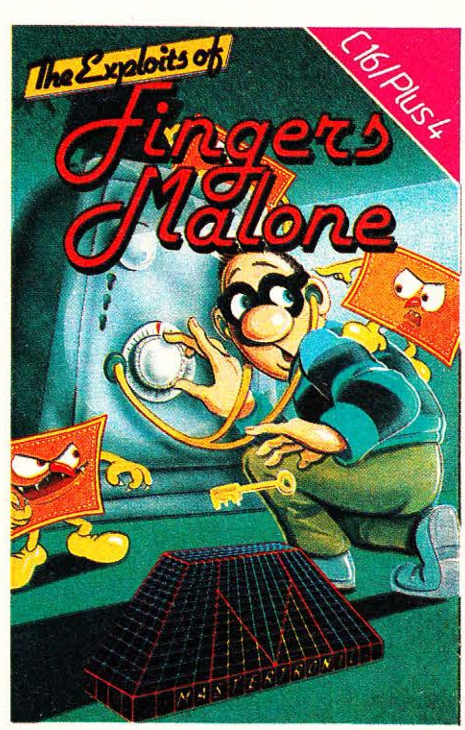
Vegas Jackpot

BMX-RACERS

Instruktionen: Deutsch.
Auf einem Kurs, welcher von oben dargestellt wird, muß das BMX-Rad gesteuert werden. Hierbei müssen Energiepunkte eingesammelt und Markierungsfähnchen überfahren werden. Durch ein Energie-(=Zeit)limit ist es relativ schwierig, ins Ziel zu gelangen.
Hier hindern noch zusätzlich die zu hohe Geschwindigkeit und die mangelnde Lenkpräzision. Die absolut mißlungene Grafik (das angebliche BMX-Rad ist ein hingeklatschter Fleck) macht das Spiel auch nicht schöner. Leider ist der Schwierigkeitsgrad nicht einstellbar, so daß der Spaß bald vergangen ist.
Nicht empfehlenswert



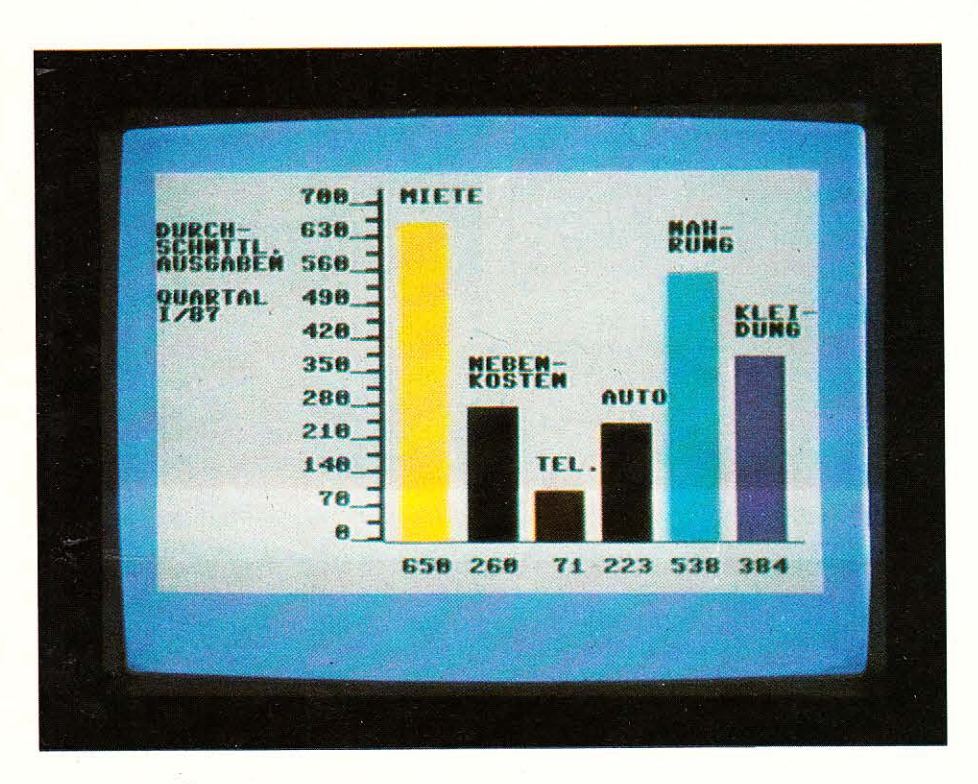
Spectipede



Finger's Malone

CALC-PLUS: KOMFORTABEL UND PREISWERT

Für weniger als vierzig Mark wird ein Tabellenkalkulationsprogramm als Modul für den Plus 4 angeboten: Calc Plus. COMMODORE WELT nahm es für Sie unter die Lupe.



„Die anspruchsvolle, jedoch einfach zu handhabende Tabellenkalkulation. Denjenigen, die eine anspruchsvolle, einfach zu handhabende Tabellenkalkulation benötigen, stellen wir Calc/Plus vor, ein ideales Kalkulationswerkzeug, welches absolut keine Programmierkenntnisse voraussetzt.

Mehrfarbige Balkengrafik kann entweder zeilen- oder spaltenweise verwandt werden. Das Programm bietet verschiedene zusätzliche Möglichkeiten wie Formulardruck, formatiertes und direktes Drucken, einen integrierten Hilfetext, änderbare Bildschirmfarben, manuelle und automatische Neuberechnungsarten, Titelsetzen und vieles mehr. Viele Disketten- und Kassettenkommandos sorgen für zusätzlichen Komfort.

Sie haben nun solch wertvolle Dinge wie automatische/manuelle Neuberechnung, variable Spaltenbreite, Kopieren, Verschieben und lokales sowie globales Formatieren fest im Griff. Und dies mit einer Kapazität von 254 Zeilen a 63 Spalten.“

Mit diesen Worten auf der Verpackung, alles natürlich in Englisch, stellt sich dieses Softwareprodukt vor. Im Innern der Verpackung finden wir ein ebenfalls englisches Handbuch und ein Steckmodul, welches in den Expansionsport unseres Computers einzuschieben ist, bei ausgeschalteter Stromversorgung versteht sich.

Beim Wiedereinschalten erscheint neben unserer gewohnten Einschaltmeldung zusätzlich CALC/PLUS ON KEY F2. Wir drücken nun diese Taste, dann RETURN und nach kurzem Aufblinken eines Copyrightkopfes von Commodore Business Machines haben wir das Calc/Plus Spreadsheet vor Augen, 21 Zeilen a 4 Spalten mit einer Breite von je 9 Zeichen.

Wer gewohnt ist, englische Handbücher zu lesen, wird nun kaum Probleme haben. Im Stich gelassen mag sich mancher andere vorkommen.

TABELLEN SIND SCHNELL MIT CALC PLUS ERSTELLT

Das erste, was wir probieren, sind die Cursortasten. Betätigen wir CURSOR-RIGHT, so wandert der in Feld A1 liegende invers dargestellte Balken um eine Spalte nach rechts. Bei fortlaufender Betätigung erreicht er schließlich den rechten Bildrand. Nunmehr verbleibt er in dieser Position, jedoch, wie die Spaltenkennung zeigt, scrollt der Bildschirm nach links, bis endlich bei Spalte 8K, der 63. Spalte, die rechte Grenze des Spreadsheets erreicht ist. Analog hierzu funktionieren die anderen Cursortasten. Die untere Grenze liegt bei Zeile 254, wie wir leicht feststellen können.

Ein wenig unkomfortabel ist es, so weite Entfernungen mit den Cursortasten zurückzulegen. Vielleicht kommen wir etwas schneller wieder zum Tabellenanfang, wenn wir uns nur recht zu helfen wüßten. Ein Druck auf die HELP-Taste bringt es an den Tag: F2 Go to cell, steht hier zu lesen. Nach dreimaligem RETURN-Druck können wir den Hilfetext verlassen und befinden uns wieder im vertrauten Spreadsheet. Wir drücken die Taste F2, geben A1 ein und schicken mit RETURN ab. Ehe wir uns recht versehen, befinden wir uns bereits in der richtigen Zelle, vorbildlich schnell, und das ist noch viel zu wenig gesagt.

Die Sache schaut bereits sehr verheißungsvoll aus. Wir wandern mit dem Cursor jetzt mal etwa in die Bildschirmmitte und drücken die HOME-Taste. Der Cursor befindet sich daraufhin im linken oberen Feld des Bildschirmes. Die CLEAR-Taste funktioniert glücklicherweise nicht, denn sonst könnte sie uns bei versehentlichem Drücken erheblichen Ärger bereiten.

Genug der Cursorbewegungen. Wir versuchen uns an Einträgen. Zahlen bereiten keine Probleme. Zur Eingabe von Text ist zuvor die SPACE-Taste zu drücken, dann erscheint links oben statt der Kennung „Value" das Wort „Label". Löschen wir nun die eventuell bereits getätigten Eingaben, indem wir mit dem Cursor auf die entsprechenden Felder gehen und die Taste F3 versuchen. Es geht auch anders, aber hierzu kommen wir später.

An einem Beispiel erarbeiten wir uns nun das Handling.

Wir gehen mit dem Cursor nach A2 und schreiben: „Ausgaben" danach drücken wir CURSOR-RIGHT und so fahren wir fort mit „Übersicht", „Quartal", „I/87", die Anführungszeichen natürlich nicht mitschreiben, sondern nur den Text dazwischen. In die Zellen B4-E4 schreiben wir „Januar", „Februar", „Maerz" und „Schnitt". In die Zellen A6-A11 „Miete", „Nebenk.", „Telefon", „Auto", „Nahrung" und „Kleidung" und schließlich in Zelle A13 „Summe". Dann füllen wir die Felder B6-D11 mit Zahlen auf. In B13 wollen wir die Summe sehen, jedoch ohne, daß wir sie

KOMFORTABLE MENÜFÜHRUNG MIT VIELEN MÖGLICHKEITEN

vorher mit dem Taschenrechner ausaddiert haben. Hierzu haben wir zwei Möglichkeiten. Entweder wir schreiben in dieses Feld die Formel B6+B7+B8+B9+B10+B11, was hier noch leicht machbar ist, oder wir verwenden die Summenformel: §SUM(B6:B11). Letzteres empfiehlt sich bei längeren Summen. Es sollte der Wert 2037 erscheinen. Wenn wir in der also berechneten Spalte irgendeinen Wert verändern, macht sich das, ohne daß wir noch etwas dazutun müssen, sofort in der Summe bemerkbar. Wir könnten die anderen Spalten ebenso berechnen. Jedoch es geht auch komfortabler, wenn wir zuvor uns weiteres Wissen aneignen.

Drücken wir doch einmal F1. Wir gelangen dadurch in den Menümodus, und es erscheint folgender Text:

function (D,T,P,M,R,F,C,W,G, Help):

Der blinkende Cursor nach dem Doppelpunkt fordert uns zur Auswahl auf. Damit wir diese treffen können, ist wichtig zu wissen, was die Buchstaben bedeuten. Die HELP-Taste bringt es an den Tag:

D = Disk Functions
T = Tape Functions
P = Print Functions
M = Cell Movement
R = Recalculation Functions
F = Format Cell Functions
C = Color Functions
W = Workspace Functions
G = Graphic Functions

Anstatt die HELP-Taste zur Information zu benützen, ist es genauso möglich, die entsprechende Funktion anzuwählen und dann das Untermenü, in welches man daraufhin gelangt, mit der STOP-Taste wieder zu verlassen. Die für uns im Augenblick interessanten Funktionen sind diejenigen, die das Cell Movement betreffen.

Nach der Anwahl des Menümodus mit der F1-Taste drücken wir die Taste M für Move Cells und gelangen in folgendes Untermenü: Insert Delete Copy Move Replicate

Insert und Delete dienen zum Einfügen und Löschen von Zeilen und Spalten, mit Copy und Move kopieren und verschieben wir Spreadsheetbereiche. Replicate kopiert nicht die Werte der Zellen, sondern, sofern Formeln Verwendung fanden, die Formeln. Auch kann nicht wie bei Copy und Move einer Zelle genau nur ein Abbild zugewiesen werden, sondern Vervielfältigung ist möglich.

Wir wählen R für Replicate. Danach erscheint auf dem Bildschirm:

Replicate Cells
Source Range: From B13 To
Target Range: From To

Die Eintragungen können auf zweierlei Weisen erfolgen, nämlich durch direktes Tippen der Zeichen, oder wir bewegen den Cursor auf das entsprechende Feld. Am Ende sollte es aussehen wie hier:

Source Range: From B13 To B13
Target Range: From C13 To E13

Danach fragt das Programm, ob die Adressen absolut oder relativ zu nehmen sind. Da wir nicht jeweils die Summe von B6 bis B11 kopieren wollen, sondern die der entsprechenden Spalte, wählen wir jedesmal „Relativ".

Zur Berechnung des Schnitts in E6 benützen wir folgende Formel:

§RINT((B6+C6+D6)/)

§RINT (Rounded Integer) heißt, daß auf die nächstliegende ganze Zahl hin gerundet wird. Zur Vollständigkeit ist hier die Liste der zusätzlichen Operatoren:

ZUSÄTZLICHE OPERATOREN

§SUM(l.o. Zelle : r.u. Zelle)
= Aufsummierung eines Datenblockes
§ABS(...)
= Absolutbetrag
§INT(...)
= Wandlung in Integerzahl
§RINT(...)
= Integerwandlung mit Rundung
§RDOL(...)
= Runden auf zwei Dezimalstellen
§EXP(...)
= Potenz zur Basis „e" (2.7182...)
§LOG(...)
= natürlicher Logarithmus (Basis „e")
(...)%
= Teilen durch 100 (Prozentrechnung)

Dieses war auch schon die ganze Liste. Wir replizieren den Schnitt in E6 folgendermaßen:

FORMELN KÖNNEN VERVIELFÄLTIGT WERDEN

Source Range: From E6 To E6
Target Range: From F7 To E11

Unsere Tabelle ist fertig, doch wollen wir sie noch etwas verschönern.

In Zelle A5 tragen wir als Label „—" ein. Das Programm macht dieses sichtbar als „——————————".

Wir replizieren:
Source Cell: From A5 To A5
Target Cell: From B5 To E5
und
Source Cell: From A5 To A5
Target Cell: From B12 To E12

Zelle A3 versehen wir mit „=========". Die neun Zeichen müssen wir jetzt aber ausschreiben, denn „—" war ein Sonderfall, vorgesehen, um Unterstreichungen zu erleichtern. Wie die Linie zu verlängern ist, ist uns mittlerweile geläufig. Uns fehlt noch eine Zeile. Wir gehen

auf A4 mit dem Cursor und fügen sie ein durch Drücken
F1 m i r
dies heißt
Menümodus Move Insert Row

Ein Letztes bleibt noch zu tun, nämlich die Überschriften rechtsbündig zu setzen. Wir gehen auf B5 und wählen:
Menümodus Format Justify Right

Dies müssen wir leider nun für die rechts davon liegenden Felder ebenfalls tun. Hätten wir es nur bereits gewußt, bevor wir die Spaltenüberschriften verfaßten. Wir hätten einen Label nach Wahl nur in ein einziges Feld einzutragen brauchen. Mit einem Justify Right und durch geeignetes Replizieren wäre die Rechtsbündigkeit in einem Zuge auf die gewünschten Felder zu übertragen gewesen.

Doch auch so haben wir es geschafft. Wir sichern unsere Übersicht mit „Disk Save" unter Angabe des Filenamens, z.B.:
F1 d s Kosten I/87

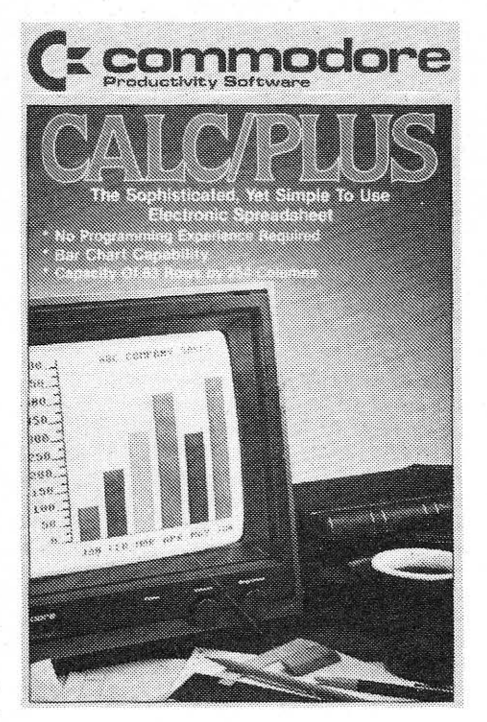
Das Ausdrucken bereitet auch keine Probleme:
F1 p d A1 E14

Print Data From A1 To E14, bringt den gewünschten Ausschnitt zu Papier.

GRAFIK AUF BILDSCHIRM UND DRUCKER MÖGLICH

Eine Tabelle können wir jedoch auch auf andere Art darstellen, nämlich grafisch. Die Grafik ist ziemlich einfach zu bedienen. Sie besitzt dafür auch nicht die Möglichkeiten, die man in einem Programm findet, welches zwischen 1000 und 2000 DM kostet.

Nehmen wir z.B. Lotus 1 2 3. In einem solchen Programm, welches nur auf einem PC läuft, haben wir z.B. die Möglichkeiten, verschiedene Typen von Diagrammen zu erstellen, wie Punkt-, Linien-, Balken- und Kreisdiagramme. Die definierten Diagramme können mit Überschriften und sonstigen Titeln versehen werden. Man kann beliebig viele dieser Diagramme wiederaufrufbar speichern, solange der Hauptspeicherplatz dieses zuläßt. Es können mehrere Spalten, nämlich sechs an der Zahl, mit beliebig langen Zeilen ausgewertet werden.



Die Möglichkeiten unseres Calc/Plus nehmen sich demgegenüber schon etwas bescheiden aus. Wählen wir „Grafik", so müssen wir uns daraufhin für die spalten- oder die zeilenweise Darstellung entscheiden. Wir brauchen nur noch die obere und die untere Grenze des Wertebereiches anzugeben und schon ist die Grafik auf dem Bildschirm sichtbar, allerdings nur als Balkengrafik mit sechs Balken, falls alle Werte innerhalb des Wertebereiches liegen bzw. es sich außerdem überhaupt um Werte (values), und nicht um leere Zellen bzw. um Text (labels) handelt.

Ein paar Bonbons erwarten uns, mit denen wir nicht gerechnet haben. Wir können den Bildschirm nach Belieben mit Text versehen, wenn wir die Taste „e" (edit) drücken. Die Texteingabe wird mit RETURN abgeschlossen. Im Grafikmodus wieder angelangt, haben wir nicht nur die Möglichkeit, denselben durch ein erneutes RETURN zu verlassen. Eine andere Taste verhilft dem nur allzu vergänglichen Diagramm auf dem Bildschirm zu dauerhafterer Daseinsweise. Die Taste „p" nämlich veranlaßt die Ausgabe des Bildschirmes auf den Drucker (Hardcopy).

Für die Darstellung einer anderen Grafik ist es nicht unbedingt nötig, daß wir den Grafikmodus verlassen. Wollen wir keine Veränderung des Wertebereiches oder der Wahl, ob die Darstellung eine spalten- oder zeilenweise sei, vornehmen, und liegt die neue gewünschte Zeile oder Spalte in erreichbarer Nähe, auch blind mit dem Cursor zu finden, so können wir durch Gebrauch der Cursor-Tasten ebenso diese Grafik auf den Bildschirm bekommen.

FAZIT: AUSGEZEICHNETES PREIS/LEISTUNGSVERHÄLTNIS

Calc/Plus bietet für den Preis von 39 DM erstaunlich viel. Über die Möglichkeiten konnten Sie sich anhand unserer bisherigen Informationen wohl bereits selbst ein Bild machen. Wir können an dieser Stelle nur noch einmal betonen, daß die Arbeit mit Calc/Plus sich als sehr komfortabel erwies. Als Steckmodul erübrigt sich die sonst übliche Ladezeit. Die Menüs sind übersichtlich gestaltet. Vor allem ist die überaus hohe Ablaufgeschwindigkeit des Programmes hervorzuheben. Damit macht das Arbeiten mit Calc/Plus so richtig Freude, was man von manch anderer Tabellenkalkulation wirklich nicht behaupten kann. Wollen wir hoffen, daß in Zukunft noch mehr solch guter, preiswerter Software auf den Markt kommt. *(Alfons Mittelmeyer)*

TEXT 16: WIRKLICH KOMPLETT?

Billig, aber viele Wünsche offenlassend: Text 16. Wir testeten für Sie.

Wer nach dem Öffnen der Verpakkung auf den zwei Umschlaginnenseiten die ganze Bedienungsanleitung abgedruckt findet, findet eher das Gegenteil: Eine einfache Textverarbeitung, die viele Wünsche offenläßt. Dies muß aber nicht unbedingt ein Manko sein.

Durch ein wirklich komplexes Textverarbeitungsprogramm mit dickem Handbuch, voll mit einer Überfülle von Möglichkeiten und Informationen, würde der Computerneuling wahrscheinlich ziemlich verwirrt und frustriert dieses wieder beiseitelegen und sein Vorhaben, einen Brief mit dem Computer zu schreiben, erst einmal auf einen späteren Zeitpunkt vertagen.

Wie groß ist dagegen seine Freude, wenn er, kaum hat er seinen Computer ausgepackt und ein wenig darauf herumgespielt, bereits in einem damit geschriebenen Brief von seiner neuesten Errungenschaft berichten kann.

Ob ihm dieses mit dem vorliegenden Programm auch so ohne weiteres möglich ist, soll der Gegenstand unserer Untersuchung sein.

Das Programm, welches uns auf Kassette vorliegt, wird gestartet mit:
LOAD " ",1,1

Nach dem Drücken der RETURN-Taste beginnt eine Folge von Ladevorgängen, bis man nach ca. 7 Minuten und bei Bandzählerstand 121 im Editiermodus landet.

Eine Zeilen- und Spaltenanzeige informiert über die augenblickliche Cursorposition. Wie sich durch Bewegung des Cursors leicht nachprüfen läßt, lassen sich 72 Zeilen a 70 Zeichen bearbeiten. Bei Erreichen des Bildschirmrandes wird automatisch gescrollt. Wird beim Schreiben das Zeilenende überschritten, erfolgt ein sogenanntes Wordwrapping, welches allerdings auch ausschaltbar ist. Hierbei wird das gerade in Bearbeitung befindliche Wort vollständig in die nächste Zeile übernommen, so daß man dem Zeilenende keine besondere Beachtung schenken muß, sondern unbeschwert einach darauf losschreiben kann.

Die INST-Taste, welche im Grunde genauso wie in Basic funktioniert, ist in der Anleitung falsch beschrieben.

Als etwas gewöhnungsbedürftig erweist sich die DEL-Taste. Nicht wie gewohnt löscht sie das vorangegangene Zeichen, sondern das, welches sich gerade unter dem Cursor befindet. Dies ist vorteilhaft, da man zum Löschen einer Zeichenkette nicht erst deren Ende aufsuchen muß. Zur Korrektur des vorangegangenen Zeichens nimmt man am besten die CURSOR-LEFT-Taste.

Sehr vermißt man die Möglichkeit, eine ganze Zeile herauszulöschen oder einzufügen, was als wirklich bedauerliches Manko dieser Textverarbeitung angesehen werden muß. Auch Sprünge an den Text- und Zeilenanfang bzw. Text- und Zeilenende sind nicht realisiert.

Doch hat die Textverarbeitung auch etwas zu bieten, was das Herz des Benutzers wieder höherschlagen läßt. Wird nämlich die HELP-Taste gedrückt, so erscheint ein Menü, welches uns über die sonstigen angebotenen Möglichkeiten informiert. Diese sind:
1: Farbeinstellung
2: Text suchen
3: Text ersetzen
4: Einfügemodus
5: Wordwrap ein/aus
6: Text löschen
7: Text speichen
8: Text laden
9: Wortsprung
a: Blockbefehle
b: Text drucken
c: Programmende

Die wichtigsten Optionen hierbei sind: Text speichern, Text laden und Text drucken.

Gespeichert und geladen werden kann erfreulicherweise sowohl auf Kassette wie auch auf Diskette. Text drucken funktionierte leider in der uns vorliegenden Version nicht.

Gut funktionierten Text suchen, Text ersetzen und Wortsprung, bei lediglich einer Seite Text wohl nicht so wichtig, jedoch sehr schön, wenn man bereits damit üben kann.

Bei den Blockbefehlen kann man zwar einen Block kopieren, jedoch vermißt man die Möglichkeit, ihn auch zu verschieben, besonders, wenn die Option Text löschen nur die Löschung von Textanfang bis Textende oder die Löschung des Gesamttextes zuläßt.

Wordwrapping wurde bereits erklärt. Die Einschaltung des Einfügemodus erlaubt die Einfügung von Buchstaben, ohne daß man sich vorher den Zwischenraum mit Hilfe der INST-Taste schaffen müßte. Leider hebt RETURN diesen Modus wieder auf.

Die Einstellung von Rahmen-, Hintergrund- und Zeichenfarbe ist wohl als Spielerei zu werten, besonders wenn durch Tastendruck nur jeweils die Farbnummer incrementiert werden kann, und da hierbei auch alle Helligkeitsstufen durchlaufen werden, man 8*16=128 Tastendrucke braucht, um wieder auf die Ausgangsfarbe zurückzukommen.

Weit wichtiger wäre wohl eine Druckeransteuerung gewesen, mit welcher sich verschiedene Schriftmodi wie Fettdruck, Unterstreichung etc. und auch verschiedene Schriftdichten hätten einstellen lassen.

FAZIT

Als abschließendes Urteil kann festgehalten werden, daß TEXT C16 den Ansprüchen, welche man an eine professionelle Textverarbeitung stellt, nicht genügt, daß sie aber, besondes auch durch die übersichtliche Menüführung, es dem Beginner recht einfach macht, mit ihr umzugehen, und für ihn, da sie einige Optionen, wie Textersetzen oder Textsuchen anbietet, eine recht brauchbare Einführung in Textverarbeitung darstellt.

A. Mittelmeyer

Einer der Vorzüge der C16-Reihe (C16/C116 und PLUS4) ist zweifelsohne der eingebaute Maschinensprache-Monitor TEDMON.
Im Anhang des Handbuches (Seite 218–225) sind die einzelnen Befehle erklärt. Der Anfänger benötigt aber doch noch einige Tips und Hinweise, wie er ihn sinnvoll verwenden kann. Dieser Artikel soll vor allem dazu dienen, einige Möglichkeiten aufzuzeigen, um dem Anfänger die Scheu vor TEDMON zu nehmen.

TEDMOND FÜR BASIC-PROGRAMMIERER

Sicher denken viele, daß sie ja vorerst mit BASIC vollauf beschäftigt sind und Maschinensprache etwas ist, das nur Freaks verstehen. Diese sollten unbedingt weiterlesen.
Man kann den TEDMON auch von BASIC aus nutzen. Will man vorhandene Maschinenroutinen in sein Programm einbauen, so kann man sie wie gewohnt mit:
LOAD "name",8,1 (RETURN)
einladen. Dabei verändern sich aber die Speicherstellen 43–46, die BASIC-Programm-Anfang und -Ende anzeigen. Mit dem L-Befehl des Monitors kann man dies umgehen.
Zunächst muß man den TEDMON mit
MONITOR (RETURN)
(oder mO als Abkürzung) aufrufen. Die Registerinhalte, welche dann angezeigt werden, interessieren uns hier nicht. Mit
L"name",8 (RETURN)
(mit ,1 bei Kassette) wird das Maschinenprogramm geladen und nach
X (RETURN)
befindet man sich wieder im BASIC, ohne daß die Speicherstellen 43–46 verändert sind. Das ganze kann man auch vom BASIC-Programm aus machen. Dazu muß man sich der Tastaturpuffermethode bedienen. Zuerst wird der Text, den man beim Laden von Hand eingeben würde, mit CHAR-Anweisungen genau an den Stellen ausgegeben, wo man ihn in obigem Beispiel auch geschrieben hat. Dann wird der Cursor auf dem ersten Befehl (MONITOR) plaziert und in den Tastaturpuffer sooft die Zahl 13 (Codezahl für RETURN) gepoket, wie man auch von Hand RETURN drücken würde. In die Speicherstelle 239 muß man noch die Anzahl der simulierten Tastendrücke poken. Dies hört sich für den Uneingeweihten vielleicht etwas kompliziert an. Deshalb hier ein kleines Beispielprogramm, welches dies alles macht.

DER MONITORLADER

Falls Sie zu denen gehören, die überhaupt keine Maschinenroutinen benutzen (was sich am Ende dieses Artikels jedoch ändern wird), so überlegen Sie sich einmal, ob Sie nicht auf diese Weise HIRES-Bildschirme mit einer hübschen Grafik abspeichern und mit obiger Methode im Programm einblenden wollen.
Abspeichern eines HIRES-Bildschirms:
MONITOR (RETURN)
S"HIRES",8,
1800,4000 (RETURN)
X (RETURN)

DER ASSEMBLER IM TEDMON

Vielleicht wollen Sie es doch einmal versuchen, ein kleines Maschinenprogramm mit Hilfe des Assemblers einzugeben? Es ist ganz einfach. Zunächst einmal das Assemblerlisting:
FLASH WEG (stoppt das Blinken)

```
. 05be a0 00      ldy #$00
. 05c0 b9 00 08   lda $0800,y
. 05c3 29 7f      and #$7f
. 05c5 99 00 08   sta $0800,y
. 05c8 b9 00 09   lda $0900,y
. 05cb 29 7f      and #$7f
. 05cd 99 00 09   sta $0900,y
. 05d0 b9 00 0a   lda $0a00,y
. 05d3 29 7f      and #$7f
. 05d5 29 7f      and $0a00,y
. 05d8 b9 00 0b   lda $0b00,y
. 05db 29 7f      and #$7f
. 05dd 99 00 0b   sta $0b00,y
. 05e0 c8         iny
. 05e1 c0 00      cpy #$00
. 05e3 d0 db      bne $05c0
. 05e5 60         rts
```

Das sieht auf den ersten Blick recht kompliziert aus, aber in wenigen Minuten werden Sie es verstehen.
Zunächst einmal müssen die sogenannten Mnemoniks, die Klartext-Assemblerbefehle erklärt werden. Bei einem Maschinenprogramm läuft praktisch alles über die Prozessorregister A, Y und Y. Am besten geben Sie mal das obige Assemblerprogramm ein, und nebenher lesen Sie die Erklärungen. Zum Eingeben eines Assemblerbefehls kann man an den Anfang der Zeile ein a oder einen Punkt schreiben:
. 05be ldy #$00
Nach dem Punkt kommt zuerst die Hexadezimal-Adresse, bei der das Maschinenprogramm beginnen soll. 'ldy #$00' bedeutet, daß die Hex-Zahl $00 in das Y-Register geladen werden soll. Die nächste Adresse muß nicht mehr geschrieben werden, die wird automatisch vorgegeben.
. 05c0 lda $0800,y
'lda $0800,y': Der Wert der Speicherstelle $0800+ (Inhalt des Y-Registers) wird in das A-Register geladen (angenommen, im Y-Register steht eine 5, so wird der Wert aus der Speicherstelle $0805 genommen).
. 05c3 and #$7f
Beim AND-Befehl wird immer der Inhalt des A-Registers mit der Zahl hinter AND 'undiert', das heißt, daß beim Ergebnis nur die Bits ungleich Null sind, die in beiden Zahlen ungleich Null sind. Durch die AND-Verknüpfung mit $7f=127 wird das 7. Bit der Zahl im A-Register gelöscht. Das Ergebnis

DER MONIT
DAS
UNBEK
WES

```
MONITOR
      PC   SR AC XR YR
;  FBØØØ ØØ ØØ ØØ ØØ

>ØØØ1B Ø1 FD FE ØØ
>ØØØ23 ØØ Ø6 Ø2 ØØ
>ØØØ2B ØØ ØØ Ø1 1C
>ØØØ33 ØØ Ø4 ØØ FF
>ØØØ3B ØØ FF Ø1 Ø2
>ØØØ43 ØØ 1C ØØ ØØ
>ØØØ4B Ø4 Ø4 ØØ ØØ
>ØØØ53 ØØ ØØ ØØ 4C
>ØØØ5B ØØ ØØ ØØ ØØ
>ØØØ63 Ø1 FD FE 63
>ØØØ6B AØ ØØ ØØ ØØ
>ØØØ73 ØØ ØØ ØØ ØØ

>ØØØ7B ØØ ØØ FF Ø9
>ØØØ83 ØØ Ø1 Ø2 ØD
>ØØØ8B ØØ ØØ ØØ ØØ
>ØØØ93 ØØ AA 2Ø ØØ
>ØØØ9B ØØ ØØ CØ ØØ
>ØØØA3 55 ØØ ØØ ØØ
>ØØØAB ØØ ØØ ØB ØØ
>ØØØB3 ØB ØØ ØØ ØØ
>ØØØBB Ø1 ØØ ØØ ØØ
>ØØØC3 73 EØ ØØ ØØ
>ØØØCB ØD 8Ø FA 1Ø
```

Monitor-Hardcopy

steht nachher wieder im A-Register.
. 05c5 sta $0800,y
Jetzt wird der Wert aus dem A-Register in die

OR-

ANNTE

EN

```
000 00 00:
000 00 00:
004 00 04:
FFE 00 FF:
000 00 00:
000 00 00:
000 00 00:
E85 00 00:
003 00 00:
000 00 85:
000 00 00:
000 29 01:

000 00 00:
000 00 00:
000 FF 80:
001 00 04:
000 0E 50:
000 00 00:
000 00 00:
004 FF 04:
300 01 00:
000 0C 00:
E00 00 00:
```

Speicherstelle $0800+ (Inhalt des Y-Registers) geschrieben.
So, die Hauptarbeit ist erledigt. Wie man sieht, ist der größte Teil vom Rest eine Wiederholung mit leichten Änderungen. Fahren Sie jetzt einfach hoch mit dem Cursor auf 05c0 und ändern es in 05c8. Dann muß noch in der gleichen Zeile 0800 in d0900 geändert werden. Nach 2mal RETURN ändern Sie wieder 0800 in 0900 und drücken RETURN.
Wenn Sie jetzt den Disassembler-Befehl
d (RETURN)
eingeben, sehen Sie schon den größten Teil des Programms. Nun haben Sie ja schon etwas Erfahrung im Überschreiben und können den Trick noch 2mal anwenden, bis Sie bei 05e0 angelangt sind.
Geben Sie nun noch ein:

```
. 05e0 iny
. 05e1 cpy #$00
. 05e3 bne$ 05c0
. 05e5 rts
```

INY (increase y) erhöht den Inhalt des Y-Registers um 1, CPY (compare y) vergleicht ihn dann mit $00 und BNE (Branch if not equal) springt bei Nichterfüllung nach $05c0 zurück. Ist der Inhalt des Y-Registers gleich Null, so wird RTS (return from subroutine) ausgeführt und das Maschinenprogramm beendet. Der aufmerksame Leser wundert sich nun vielleicht, wie das Y-Register 0 werden soll, wenn immer 1 dazugezählt wird? Der größte Inhalt einer Speicherstelle eines 8-Bit-Computers ist 255 = 2 hoch 8 –1. Wird dazu nochmals 1 gezählt, so ergibt es wieder Null und in einer anderen Speicherstelle wird ein Bit gesetzt, welches anzeigt, daß hier ein Überlauf stattgefunden hat.

ENDLICH SCHLUSS MIT DEM NERVÖSEN BLINKEN BEI HELP

Nun soll auch endlich erklärt werden, was dieses Programm leistet. Sie kennen ja sicher die Funktion der HELP-Taste.
Drückt man diese nach einem Programmfehler, so wird die fehlerhafte Stelle gelistet und blinkt, was einen auf die Dauer ziemlich nervös machen kann und beim Korrigieren ablenkt.
Mit diesem kleinen Maschinenprogramm ist damit Schluß. In den Speicherstellen von $0800 (dez. 2048) bis $0BFF (dez. 3071) befindet sich der Farbspeicher. Für jedes Zeichen auf dem Bildschirm kann die Farbe (16 Möglichkeiten = 4 Bits), die Luminanz (7 Möglichkeiten = 3 Bits) und FLASH ON oder FLASH OFF (2 Möglichkeiten = 1 Bit) festgelegt werden. Da jedes Byte 8 Bit hat, kann alles in diesem einen KBYte untergebracht werden. Hierbei wird der FLASH-Modus durch ein gesetztes 7. Bit angezeigt. Wird der Speicherinhalt daher mit 127 undiert, so wird das Blinken abgestellt. Das Maschinenprogramm macht also das gleiche, wie dieses kleine BASIC-Programm:

```
10 for i=2048 to
3071:pokei,peek(i)and
127):next
```

Nur natürlich viel schneller. Der Platz für das Maschinenprogramm wurde am Ende des Funktionstastenspeichers gewählt, wo normalerweise diese 40 Byte leicht untergebracht werden können. Genaueres darüber steht in dem Artikel über TIPS & TRICKS ZU KEY.
Um Übung zu bekommen, sollten Sie jetzt noch folgendes Assemblerprogramm abtippen. Dies macht das Gleiche mit dem 7. Bit im Textspeicher, wodurch alle reversen Zeichen normal werden. Mit ORA #$80 kann man übrigens genau den gegenteiligen Effekt erzielen. Wenn Sie die beiden Assemblerlistings vergleichen, so werden Sie große Ähnlichkeiten feststellen, die beim Eingeben bzw. Umwandeln des Ersten in das Zweite sehr hilfreich sind. Versuchen Sie dann mal durch Veränderungen eigene Programme zu schreiben. Zum Beispiel kann man damit einen Bildschirminhalt verschieben, indem man die AND-Befehle wegläßt und hinter die STA-Befehle eine andere Adresse schreibt. Oder man läßt die STA- und die AND-Befehle weg und schreibt die LDA-Befehle so: LDA #$01. Jetzt wird der Bildschirm mit lauter 'A' gefüllt.

REV WEG (macht revers zu normal)

```
. 065e a0 00     ldy  $00
. 0660 b9 00 0c lda$0c00,y
. 0663 29 7f     and #$7f
. 0665 99 00 0c sta $0c00,y
. 0668 b9 00 0d lda$ 0d00,y
. 066b 29 7f     and #$7f
. 066d 99 00 0d sta $0d00,y
. 0670 b9 00 0e lda $0e00,y
. 0673 29 zf     and #$7f
. 0675 99 00 0e sta $0e00,y
. 0678 b9 00 0f lda $0f00,y
. 067b 29 7f     and #$7f
. 067d 99 00 0f sta $0f00,y
. 0680 c8        iny
. 0681 c0 00     cpy #$00
. 0683 d0 db     bne$ 0660
. 0685 60        rts
```

Das Programm wird im TEDMON mit s"rev weg", 8,065e,0686 (RETURN) abgespeichert und von BASIC aus mit SYS1630 gestartet.

ASSEMBLERLISTING AUF DEN DRUCKER

Will man ein Assemblerlisting ausdrucken, so muß man wie folgt vorgehen:

```
open1,4:cmd1      (RETURN)
monitor           (RETURN)
d 065e 0686       (RETURN)
x                 (RETURN)
print #1:close1   (RETURN)
```

Das Gleiche geht natürlich auch mit dem M-Befehl.
Platz für Maschinenprogramme:
Hier ein paar Speicherbereiche, die beim C16/116 für Maschinenprogramme benutzbar sind (wenn die angegebenen Funktionen nicht benutzt werden):
RS232 Puffer
03f7–436
Funktionstastenspeicher
055f–05e–06ee
Anwenderbereich
00d8–00e8
Kassettenpuffer
0333–03f3

Gewusst wie:

LISTSCHUTZ

Ein leidiges Thema für Programmierer ist der Schutz vor unbefugter Vervielfältigung. Damit Sie wenigstens halbwegs sicher vor den Raubkopierern sind, zeigen wir Ihnen hier, wie Programme geschützt werden können.

Wird einem Farb- und Steuercode zwischen 128 und 160 ein 141 vorgestellt, so wird zunächst das SHIFT/RETURN ausgeführt und die darauffolgenden Steuerzeichen stehen alleine auf der nächsten Zeile. Und prompt werden sie auch ausgeführt, was ohne 141 unmöglich ist. Es geht auch mit dem Code 13 (RETURN).

Das wollen wir doch mal näher betrachten:
Geben Sie bitte ein:
1PRINT":(CLEAR) (6*DOWN) (6*RIGHT) (BLACK)HALLO(BLUE) (6*UP)"

Anstelle der Klammerausdrücke muß die entsprechende Taste gedrückt werden. Zum Beispiel für (CLEAR) drückt man SHIFT/CLR HOME. Nun noch folgendes eingeben:
POKE 2055,141:LIST

Es steht ein schwarzes HALLO einsam am Bildschirm und sonst scheint sich nichts mehr zu tun. Die READY-Meldung und jede Eingabe ist jetzt wegen der blauen Farbe unsichtbar. Mit diesem Trick kann man also Listings verschwinden lassen oder beliebig einfärben (Zum Verschönern von Listings mit solchen und ähnlichen Methoden habe ich ein Programm (Listingchanger) geschrieben, das in der CBM-REVUE 5/85 S. 42 erschienen ist.) Der aufmerksame Leser hat sicher bemerkt, daß uns noch die Codes kleiner 32 fehlen. In Tabelle 1 sind alle Steuercodes zusammengefaßt. Besondere Beachtung verdienen noch die 13, die 19, die 20 und die 31. Das Tolle an den Codes unter 32 ist die Tatsache, daß sie auch beim Listen ausgeführt werden, 13 und 20 sogar innerhalb von Anführungszeichen. Vielleicht kennen Sie die Methode, wie man mit dem Code 20 (DEL) Zeilen oder Teile davon unsichtbar machen kann.

Hier ein Beispiel:
1 PRINT „TEST":GOTO 1:REM"TTTTTTTTTTT

Hierzu muß man (nur für die, die es noch nicht wissen, nach REM 2 mal Anführungszeichen, 1 mal DEL, 12 mal SHIFT/INST und 12 mal DEL



Diskettenknacken? Jetzt nicht mehr

drücken (diese erscheinen als in-verse T). Durch diese Manipulation ist das GOTO 1 verschwunden. Interessant ist die Kombination mit REMSHIFT/L:
1 REM
2 REM"TTTTTTT"L
(L=SHIFT/L)

Dadurch, daß ein DEL-Code mehr eingegeben wurde, als zum Löschen von REM und Zeilennummer nötig, wandert das zweite Anführungszeichen (wichtig, da sonst SHIFT /L als Grafikzeichen gelistet wird und keinen SYNTAX ERROR hervorruft) an das Ende der Zeile 1, so daß es aussieht, als wäre der Fehler in Zeile 1. Anstelle der Gänsefüßchenmethode kann man die 20 auch in vorher reservierte Speicherstellen POKEn.

Eine andere Möglichkeit, den „Gegner" zum Löschen einer falschen Zeile zu verlocken, ist der RETURN-Coe (13):
10 REM TEST*20 REM L (wieder SHIFT/L!)
das Sternchen überPOKEn wir jetzt mit POKE 2062,13. Dadurch erscheint beim Listen auf dem Bildschirm:
10 REM
20 REM
SYNTAXERROR

Der Versuch, Zeile 20 zu löschen, ist natürlich sinnlos. Dabei wird nur die tatsächliche Zeile 20 gelöscht.

Durch
0 REM (RETURN)
POKE 2054,31:POKE 2051,255:POKE2052, 255
kann man das ganze Listing blau einfärben und somit unsichtbar machen. Hierbei darf das Zurücksetzen der Zeilennummer im Programm nicht vergessen werden.

Um zu verhindern, daß ein geschütztes Programm nach irgendwelchen Änderungen, wie z.B. dem Entfernen des Copyrights, noch lauffähig bleibt, gibt es wiederum verschiedene Methoden.

Da ist zunächst eine im Programm versteckte Abfrage nach dem Inhalt bestimmter Speicherstellen, wie z.B. 45 und 46, oder FRE(0).
Zum Beispiel:
1000 if fre(0) < > xxxxs then new

Für xxxxx wird der Wert eingesetzt, den man bei fertigem Programm mit ?fre(0) erhält.

IF GOTO=END THEN AND=OR?????

Eine sehr interessante und weniger bekannte Methode ist die Verwendung von „unmöglichen" Variablennamen wie ST, END, TO, IF ... Versuchen Sie einmal, der Variablen ST den Wert 3 zuzuordnen:
1 ST = 3.?ST

Das geht normalerweise deshalb nicht, weil ST, IF ... zu den reservierten Wörtern gehören. Auch an dieser Stelle kann man den Computer „überlisten". Tippt man innerhalb eines reservierten Wortes einen geshifteten Buchstaben ein, so wird das Wort nicht erkannt und als Variablename eingestuft:
1 E(SHIFT/E)ND = +: PRINT E(SHIFT/E)ND
ergibt beim Listen:
1 END = 3:PRINT END
und mit RUN wird END als normale Variable behandelt. Wenn Sie aber mit dem Cursor auf die Zeile fahren und RETURN drücken, läuft es natürlich nicht mehr (Sehen Sie sich mal mit dem MINIMONITOR den Speicherinhalt in beiden Fällen an.). Es ist nicht schwer, sich nun mit diesen Kenntnissen einige Fallstricke durch Kombination der Möglichkeiten selbst auszudenken.

Doch Halt. Wir haben noch etwas sehr wichtiges vergessen. Was machen wir, wenn der „unbefugte Lister" einen Drucker besitzt?

Hier muß man sich ein wenig mit der Druckeransteuerung beschäftigen. Der Code 8 erzeugt am Bildschirm eine Verriegelung der Umschaltung zwischen Groß- und Kleinschrift. Beim Drucker wird dadurch in den Grafik-Modus umgeschaltet, und er erwartet eine Grafikinformation. Folgt auf die 8 eine Zahl kleiner 128, so bricht der Drucker das Listing ab. Außer durch hineinPOKEn in den entsprechenden Speicherbereich kann man den Coe 8, welcher auf dem Schirm als reverses H erscheint (nach "), durch folgende Methode einsetzen:
10 rem test" (CTRL/H)"

Sie merken die Wirkung dadurch, daß sich nach dem Listen der Groß/Kleinmodus nicht mehr einstellen läßt.

Das stoppt jeden Drucker

Leider gibt es aber Drucker, die sich um nicht druckbare Zeichen gar nicht kümmern.

Daher zeigen wir jetzt noch eine wesentlich bessere Methode, welche unseres Wissens immer funktioniert:
10 :::::PRINT ABC
PIKE 2053,0:LIST

Dadurch steht am Anfang der Basiczeile eine Null, wodurch der Computer ‚denkt', daß sie hier auch schon zuende ist. Beim Programmdurchlauf werden jedoch die vier auf die Null folgenden Zeichen übersprungen. Hier haben wir nun einen Listschutz, der (endlich) sowohl auf dem Schirm als auch auf dem Drucker funktioniert.

Und damit kommen wir zum Ende der uns bekannten Listschutzmethoden. Man kann auch noch mit Veränderungen der Basicendeadressen in 45 und 46 experimentieren. Zum Beispiel, indem man die drei Nullen am Basicende wegPOKEt und die Adresse in 45 und 46 erhöht. Im Programm müssen sie dann wieder zusammen mit der letzten Linkadresse vor der letzten Zeile wiederhergestellt werden. Die Ergebnisse dieser Methode sind aber stark vom Inhalt der Speicherzellen vor dem Laden des Programms abhängig und schienen uns nicht so empfehlenswert.

Selbermachen ist der Sinn

Last but not least:

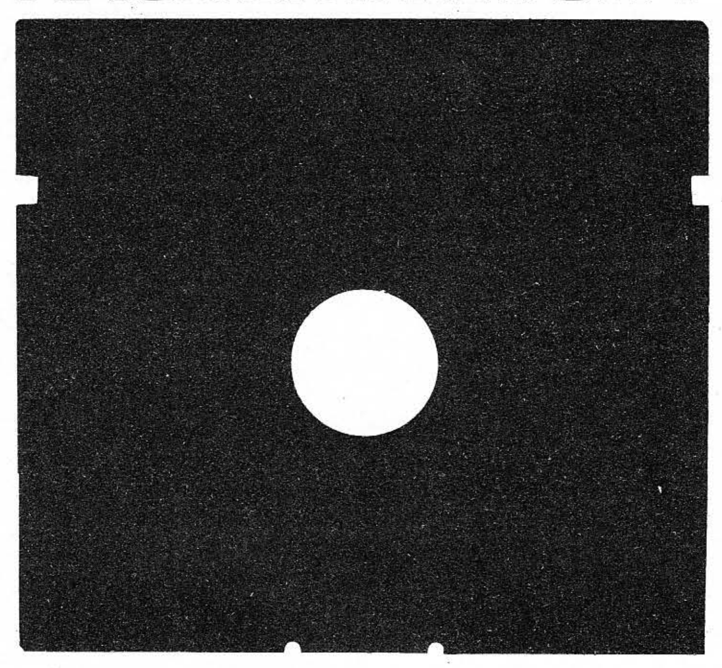
Man kann sein Programm auch noch verschlüsseln. Eine wirksame Methode dazu wurde in der CBM-REVUE 2/86 beschrieben. Hierbei wird der gesamte Speicherinhalt durch eine EOR-Verknüpfung verändert, so daß das Programm nur noch mit einem Codewort zum Laufen zu bringen ist. Zum reinen Listschutz eignet sich die Methode weniger, denn will man das Programm (nur zur Benutzung) weitergeben, so muß man auch das Paßwort preisgeben und damit kann man das Programm ja entschlüsseln. Jede Listschutzmethode ist nur solange wirksam, wie sie niemand kennt. Aber auch mit diesen, nun allen CBM-Lesern bekannten, Tricks kann man durch Kombination verschiedener Methoden einen sehr wirksamen Schutz aufbauen, der zum Knacken fast soviel Zeit und Kenntnisse erfordert, wie das Programm selbst zu schreiben. Und das ist ja der Sinn des Ganzen.

WAS IST DRAN AM DISKETTEN-KAUDERWELSCH?

Disketten, mittlerweile ein täglicher Gebrauchsgegenstand für jeden Floppy-Besitzer, überschwemmen zur Zeit in allen möglichen Variationen unter allen möglichen Markennamen den Markt.
Fast unüberschaubar scheint das riesige Angebot für den noch unerfahrenen Laien, selbst die Fortgeschrittenen staunen über die immer und immer wieder neu auftauchenden verschiedenen Diskettenmarken.
„Was nehmen? ", fragt sich da so mancher frischgebackene VC 1541-Besitzer.

NICHT JEDER USER KANN SICH AUSKENNEN

Und in der Tat:
Bei so vielen Angeboten, die auch preislich recht weit auseinanderliegen, kann ein unerfahrener Computer-User zwangsläufig nicht wissen, was für seinen Bedarf am Besten ist.
Beginnen wir einmal bei den Diskettenbezeichnungen:
Daß 5,25 Zoll die Größe der Diskette für das Homecomputerlaufwerk VC 1541 darstellt, ist noch den meisten bekannt.
Doch was bedeutet da die Bezeichnung "1D" oder "1S" oder „single sided/double density" kurz auch ss/dd genannt?
Bedeutet "single sided" etwa, daß man die Diskette nur einseitig benutzen kann; und soll "double density" eine Bezeichnung für doppelte Speicherkapazität sein?
Nur die erfahrenen User wissen, welcher Bezeichnung welche Bedeutung zukommt, doch auch die unerfahrenen Computerbenutzer können jetzt ihren Erfahrungsschatz erweitern:
"single sided" = (ss) bedeutet zwar auf Deutsch "einseitig", deshalb kann man die zweite Seite der Diskette dennoch in den meisten Fällen nach Zuhilfenahme



Eine zweite Kerbe macht aus Ihrer einseitigen eine doppelseitige Diskette

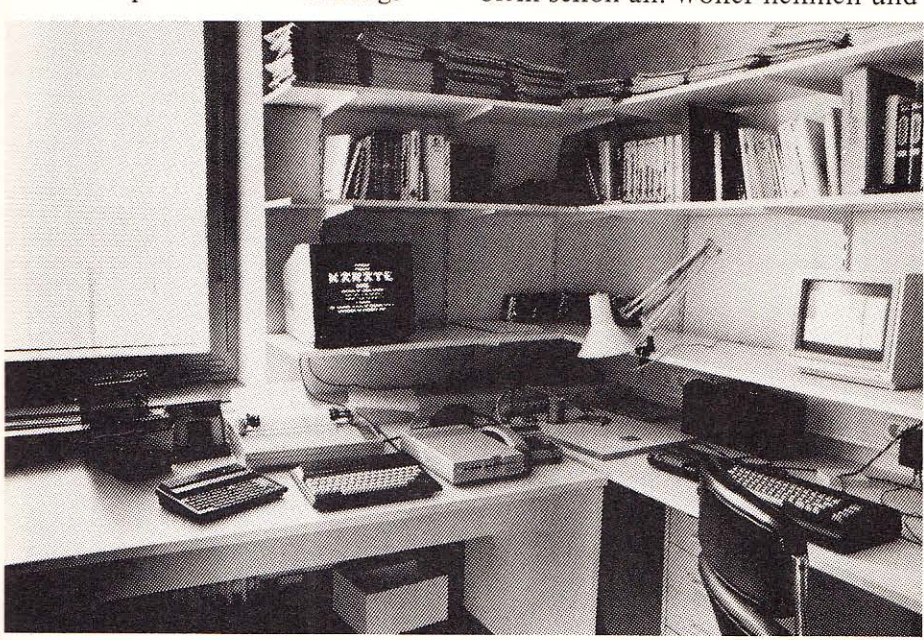
eines kleinen Tricks auch benutzen.
Der Trick besteht darin, daß man an der Diskette eine zweite Kerbe einschneidet. Wo genau, ist aus der Grafik zu ersehen.
Was bedeutet dann aber die Bezeichnung "single density" und welchen Unterschied gibt es zu einer "double sided"-Diskette?
Die Antwort darauf ist folgende:
Eine "ss" Diskette ist nur auf einer Seite geprüft; d.h. nur die normal üblich benutzbare Seite funktioniert mit Werksgarantie.
Die zweite Seite ist zwar genauso benutzbar wie die erste Seite, nur ist sie nicht getestet.
Eine "ds" Diskette wiederum ist im Prinzip dasselbe wie eine oben erwähnte "ss" Diskette, nur daß hier zusätzlich auch noch die zweite Seite auf 100%ige Funktionssicherheit überprüft wurde.
Ansonsten besteht zwischen beiden Disketten kein Unterschied, außer dem eines wesentlich höheren Preises seitens der "double sided"-Diskettenversion.
Aus diesem Grund ist vom Kauf einer "ds"-Diskette, wenn Sie über 5 Pfennig mehr kostet als der "signle sided"-Typ, abzuraten (unnötige Geldausgabe).
Manche Disketten tragen nun noch die Bezeichnung "two eye".
"Zwei Augen" heißt, daß die Diskette schon von vorneherein eine zweite Kerbe besitzt; man muß bei ihr also keine Behandlung mit Schere oder Locher wagen, so wie in Grafik I angezeigt.
Da diese Diskette aber auch meist um mehr als 20 Pfennige teurer ist als die normale "single sided"-Schwester kann ich auch hierbei nur sagen: Der Kauf einer solchen Diskette ist reine Geldverschwendung.
Kommen wir somit zu einer weiteren Bezeichnung: Der Dichte (density).
Es gibt Disketten mit der Bezeichnung "single density" = "sd" und Disketten mit der Bezeichnung "double density" = "dd".
Wiederum glauben nun manche Computerbesitzer, daß Sie auf einer "double density"-Diskette mehr Programme abspeichern können, als auf einer „single density"-Diskette.
Es wäre aber auch zu schön, wenn das klappen würde! in Wirklichkeit ist es aber leider relativ egal, ob man eine single density oder eine double density Diskette benutzt — die Speicherkapazität bleibt die gleiche!!

WAS NUN WIRKLICH KAUFEN?

Da allerdings doppelte Dichte-Disketten eine geringfügig höhere Datensicherheit gegenüber ihren einfach dichten Schwestern aufweisen und im Preis meist kein allzu großer Preisunterschied besteht, rate ich zum Kauf von double density-Datenträgern.
Die Bezeichnung „Wendediskette" ist gleichbedeutend mit "two-eye".
Fassen wir also noch einmal zusammen:
single sided = 2seitig beschichtet und benutzbar aber nur 1seitig geprüft.
double sided = 2seitig beschichtet und benutzbar und auch 2seitig geprüft.
single density = auf einfache Dichte geprüft.
double density = auf doppelte Dichte geprüft.
Eine single sided/single density-Diskette wird auch 1S genannt,
eine double sided/single density-Diskette wird auch 2 S genannt,
eine single sided/double density-Diskette wird auch 1D genannt,
eine double sided/double density-Diskette wird auch 2D genannt.
Für das VC 1541-Laufwerk sind am besten die 1D-Disketten, also single sided/double density (ss/dd) geeignet.
Sie liegen im Preis/Leistungsverhältnis recht günstig und erfüllen ihren Zweck, die Datensicherung, sehr gut.

WEM DER COMMODORE-DRUCKER NICHT PASST

Plus4, Görlitz-Interface, Mannesmann MT80. Eine abenteuerliche Kombination, werden viele sagen, obwohl sie gar nicht so ungewöhnlich ist. Der Computer (Plus/4) stellt durch seine gute Tastatur, in Verbindung mit dem Textsystem Script/Plus, eine gute und, wie ich meine, preisgünstige Lösung für ein Textverarbeitungssystem für den Heimanwender zur Verfügung. Das große Problem ist und bleibt der Drucker. Commodore Drucker sind, wie wir alle wissen, in punkto Schriftqualität und Nutzungsvielfalt nicht das Gelbe vom Ei. Schaut man sich auf dem Druckermarkt um, schreit alles nur ‚Epson'. Habe ich nicht. Ich besitze einen MT80 von Mannesmann Tally und stand nun vor dem Problem, mein Textverarbeitungssystem mit einem Drucker zu verbinden. Da der MT80 einen Centronicanschluß hat, mußte ein Interface her, das es mir ermöglichte, das Geschriebene bzw. den im Speicher stehenden Text oder Programme zu Papier zu bringen. Auf dem ständig wachsendem Markt der Schnittstellen verschiedenster Konstruktionen, die als Hardware oder Softwarelösung angeboten werden, galt es nun, das für meine Zwecke geeignete Interface auszuwählen. Eine Softwarelösung lassen der Userport und das Textmodul in Richtung Centronic nicht zu. Also muß eine Hardwarelösung her. Als geeignet schien mir das von der Firma Görlitz-Computerbau angebotene VC-Interface. Überwiegend wird es als Einbaumodul für Epson-Drucker angeboten. Aber es gibt auch eine externe Version, an die andere Druckertypen angeschlossen werden können. Da das externe Interface natürlich eine Versorgungsspannung benötigte, fing das Problem schon an. Woher nehmen und nicht stehlen. Im Gegensatz zu anderen Interface, die die benötigte Spannung dem Kassettenport entnehmen, muß hier die Spannung aus dem Drucker kommen, die an Pin 18 des Centronicsteckers zur Verfügung stehen muß. Hier ging es dann schon los. Wer hat schon einen Schaltplan seines Druckers und weiß dann schon, wo er die Spannung abgreifen kann? Hier sollte Görlitz sich etwas einfallen lassen. Wenn dann auch diese Hürde genommen ist, kann es dann richtig losgehen. Der eingebaute Selbsttest des Interface zeigt dann, was das Interface alles kann. Ein neuer Zeichensatz in allen möglichen Variationen. Auch kann der Original Commodore Zeichensatz in bis zu vierzig Variationen, in doppelter Breite und Höhe und auch Revers gedruckt werden. Beim MT80 ist mir dieses noch nicht so recht gelungen. Vielleicht ist der Drucker doch nicht so kompatibel. Im 33-seitigen Handbuch erfährt der Benutzer so ziemlich alles, was er braucht, um seinen Drucker so auszunutzen, wie er sich das erträumt hatte. Durch die softwaremäßig einstellbaren Adressen ist es möglich, mit dem Drucker alle möglichen Ausdrucke, vom Text bis zur Hardcopy, zu erstellen. Auch können je nach eingestellter Adresse alle Commodorespezifischen Steuerzeichen und Grafiksymbole ausgedruckt werden. Das Interface besitzt auch einen Linearkanal, mit dem Zeichen, ohne Umcodierung, an den Drucker gesendet werden. Da die Linearfunktion die Anwendung jeglicher Steuerbefehle verbietet, ist die Sekundäradresse auch nicht mit 'ESCAPE Space X' festzuklemmen. Der Sinn des Linearkanals liegt eben darin, eine mögliche Umcodierung zu umgehen. Wenn man, z.B. mit eigenen Programmen, hochauflösende Grafik, speziell in Maschinensprache übertragen will, wird eine Verfälschung der Übertragung vermieden.



Arbeitsplatz eines 16/P4-Redakteurs

EINWANDFREIE TEXT- UND GRAFIKWIEDERGABE

Alle im Handbuch aufgeführten Befehle zu erläutern, würde wahrscheinlich ein halbes Heft füllen. Da aber das Handbuch sehr ausführlich und übersichtlich alle Befehle, einschließlich einiger Beispiele als Listing, erklärt, wollen wir es hiermit bewenden lassen. In meiner Konfiguration Plus/4 und MT80 arbeitet das Interface einwandfrei und ordentlich, wenn man davon absieht, daß einige Druckeranweisungen nicht so hundertprozentig ausgeführt werden, was auch am Drucker liegen kann. Wiederum werden alle Befehle des Textprogramms, die für Epson-Drucker bestimmt sind, klaglos verarbeitet. Wenn man von der Suche nach einer Versorgungsspannung im Drucker absieht, die doch, da ohne Schaltplan, einige Zeit in Anspruch nimmt, kann das Interface sofort nach dem Einstecken in Betrieb genommen werden. Auch brauchte ich keine Schalter einstellen, da die Grundeinstellung wohl richtig war. Alles in allem ein Interface, das trotz seines Preises eine gute Möglichkeit bietet, seinen Drucker voll zu nutzen.

Bernd Welte

KLEIN UND FLEISSIG OKIMATE 20

Welcher Commodore-Besitzer möchte nicht gerne die Bilder, die er auf seinem Computer erstellt hat, irgendwie verewigen? Bisher gab es dafür vor allem eine Möglichkeit: Mit dem Fotoapparat ein Bildschirmfoto zu machen. Wir haben ihn am Amiga getestet, er paßt auch für die „Kleinen".

Schöner wäre es, wenn man die Bilder ausdrukken könnte; eine Möglichkeit, die viele Grafikprogramme auch vorsehen. Aber was ist so ein Schwarzweiß-Ausdruck schon gegen die Farbenpracht eines Commodores? Also muß ein Farbdrucker her. Und da bietet sich der Okimate 20 geradezu an. Mit seinem Preis von unter 800 DM ist er für nahezu jeden erschwinglich.

Den Okimate 20 gibt es in drei verschiedenen Ausführungen: Mit Centronics- oder RS-232-Schnittstelle oder IBM-kompatibel. Und genau letztere Version benötigen wir, denn damit haben wir zwar noch keinen richtigen Amiga-Drucker, aber wenigstens die Umlaute liegen an den richtigen Stellen, so daß wir ihn auch für die Textverarbeitung einsetzen können.

THERMODRUCKER

Der Okimate 20 ist ein sogenannter Thermodrukker. Das heißt, daß die Schriftzeichen durch Wärmeeinwirkung zu Papier gebracht werden. Dazu benötigt man entweder spezielles wärmeempfindliches Druckerpapier, oder man benutzt die Thermo-Farbbänder von Oki, mit deren Hilfe man dann auf jedes beliebige Papier und sogar auf Klarsichtfolien drukken kann.

Weitere Vorteile: Der Okimate 20 ist leise. Nichts hämmert auf dem Papier herum und weckt die Nachbarn, man hört nur das Hin- und Herfahren des Druckkopfes und eventuell ein leises Zirpen vom Farbband.

SCHÖNSCHRIFT

Bestechend am Okimate 20 ist sein Schönschrift-Modus, in dem die Buchstaben aus 14x18 Punkten zusammengesetzt werden. Da die Farbe durch Wärmeentwicklung auf das Papier „aufgeschmolzen" wird, fließen die einzelnen Matrixpunkte zusätzlich zusammen, so daß sich insgesamt ein sehr geschlossenes Schriftbild ergibt.

Allerdings ist die Schriftqualität sehr von der Art



des verwendeten Papiers abhängig: Es sollte eine absolut glatte Oberfläche haben. Bei rauhem Papier hat das Farbband keinen vernünftigen Kontakt zur Papieroberfläche, und die Schrift wird verwaschen und unscharf.

SCHNELLDRUCK

Wenn man keine Schönschrift benötigt, kann man den Okimate 20 auch in einen Schnelldruck-Modus schalten (per DIP-Schalter oder per ESC-Sequenz), in dem er 80 Zeichen/Sekunde druckt. Das ist zugegebenermaßen nicht atemberaubend schnell, aber wenn man die Geschwindigkeit noch höher setzen würde, hätte die Farbe auf dem Farbband gar keine Zeit mehr, zu schmelzen.
In beiden Schriftqualitäten beherrscht der Okimate 20 doppelte Druckbreite, Unterstreichen und Kursivschrift; im Schnelldruck-Modus kann er auch noch mit variablen Zeichenbreiten von 10, 12 oder 17 Zeichen/Zoll betrieben werden. Schließlich lassen sich auch noch selbsterstellte Zeichensätze laden, auch das in der Matrix von 14x18 Punkten.

GRAFIK

Zum Ausdrucken von Grafiken dienen die Bit-Image-Betriebsarten: Einfache Dichte, doppelte Dichte, doppelte Dichte mit doppelter Geschwindigkeit und vierfache Dichte, dazu ein besonders hoch auflösender Modus mit 24 Punkten übereinander statt der sonst üblichen 8 Punkte. Wechselt man das Farbband aus, kann man mit dem Okimate 20 auch farbig drucken. So einfach, wie sich das anhört, ist das allerdings nicht; es erfordert schon einige Überlegung.

FARBE

Das Farbband besteht nämlich aus kurzen Abschnitten, die verschiedene Farben haben und regelmäßig hintereinander folgen: Gelb, Magenta, Cyan, Gelb, Magenta, Cyan und so weiter.
Um nun eine Zeile mit farbigem Text zu drucken, gibt man dem Drucker zunächst mit
LPRINT CHR$(27) CHR$(25);
den Befehl, den Anfang so einer Farb-Dreiergruppe zu suchen. Die nächsten drei LPRINT-Befehle drucken dann jeweils in einer der drei Farben. Damit sie in einer Zeile erscheinen, dürfen die beiden ersten LPRINTs keinen Zeilenvorschub machen, sondern müssen mit CHR$(13); abgeschlossen werden, so daß der Druckkopf zum Zeilenanfang zurückgeht. Erst der dritte LPRINT-Befehl darf das Papier weitertransportieren.
Um verschiedene Farbtöne zu erzeugen, kann man mehrere Farben übereinanderdrucken, oder man druckt mit der Bit-Image-Grafik ein Raster, so daß sich noch weitere Zwischenwerte in den Farben erzielen lassen. Oki gibt im Druckerhandbuch selbst zu, daß dieses Verfahren etwas aufwendig ist, gibt aber zwei Beispiele für farbigen Text und farbige Grafik. Dennoch sollte man sich keine Illusionen machen, daß man nur den Drucker einzuschalten braucht und dann gleich Bilder wie die zu unserem Bericht zu Papier bringen kann. Wenn man Oki dazu überreden könnte, dem Okimate 20 gleich eine Cassette oder Diskette mit einem passenden Treiberprogramm beizulegen, wäre den Amiga-Besitzern schon viel geholfen.

PAPIERTRANSPORT

Ein Problem beim Ausdrucken von Grafiken ist immer der Papiertransport: Schiebt man das Papier zu weit, erhält man weiße Streifen zwischen den Druckzeilen, ist der Vorschub nicht weit genug, überlappen sich die Punkte und man erhält einen dunklen Strich.
Hier hat uns der Okimate 20 etwas Kummer bereitet, denn der Transport ist nicht gleichmäßig. Jeder dritte Papiervorschub ist etwas weiter als die anderen, was sicher auf Ungenauigkeiten in der Mechanik zurückzuführen ist.
Dennoch müssen wir die Druckergebnisse als "beeindruckend" bezeichnen.

DIE ANLEITUNG

Das Anleitungsbuch zum Okimate 20 umfaßt 91 Seiten und ist damit eher "knapp" zu nennen, aber es ist verständlich und in fehlerfreiem Deutsch.



Trotz seines niedrigen Preises kann der Okimate 20 mit seiner Druckqualität durchaus überzeugen. Durch die wachsartige Druckoberfläche kommt das Druckbild dem Original auf dem Bildschirm sehr nahe.



DAS PAPIER

Kommen wir noch einmal kurz zum Papier. Der Okimate 20 schluckt sowohl Endlospapier mit Traktorlochung als auch Einzelblätter. Etwas lästig fanden wir es, daß man den Papierende-Sensor nicht abschalten kann. Deshalb kann man Einzelblätter nicht bis zum Ende bedrucken, wenn man nicht "trickst" und ein zweites Blatt hinterherschiebt. Und gerade, um eine hochwertige Druckqualität zu erzielen, ist man auf Einzelblätter angewiesen, denn Endlospapier in ausreichender Qualität ist uns bislang noch nicht begegnet.

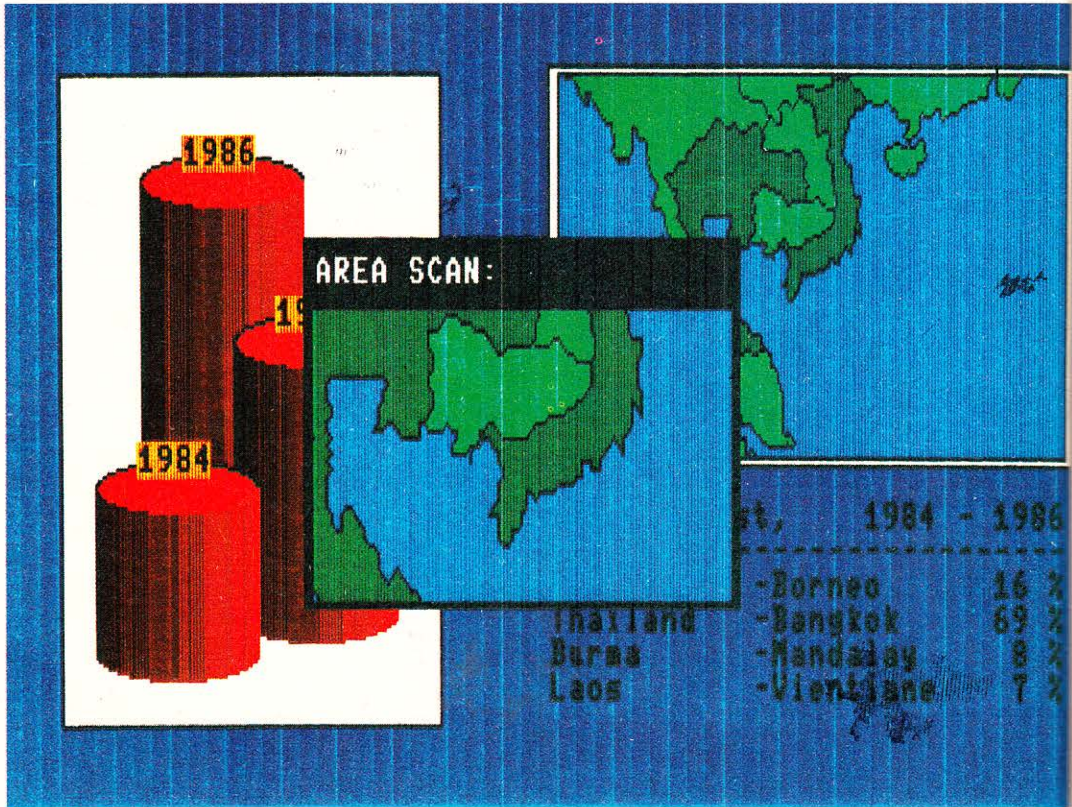
DIE FARBBÄNDER

Weil bei dem Druckvorgang die Farbe komplett vom Farbband auf das Papier übertragen wird, kann jedes Farbband nur einmal benutzt werden, danach ist es reif für den Mülleimer. Oki gibt die Lebensdauer eines Farbbandes mit 120 000 Zeichen oder 75 Seiten an, bei Farbdruck entsprechend nur ein Drittel, weil jede Zeile dreimal überdruckt werden muß. Man kann bei farbigen Bildschirm-Hardcopies ungefähr von einer bis zwei DM pro Seite ausgehen; das ist zwar nicht unbedingt spottbillig zu nennen, aber immer noch billiger, als ein Bildschirmfoto auf DIN A4-Format vergrößern zu lassen.

FAZIT

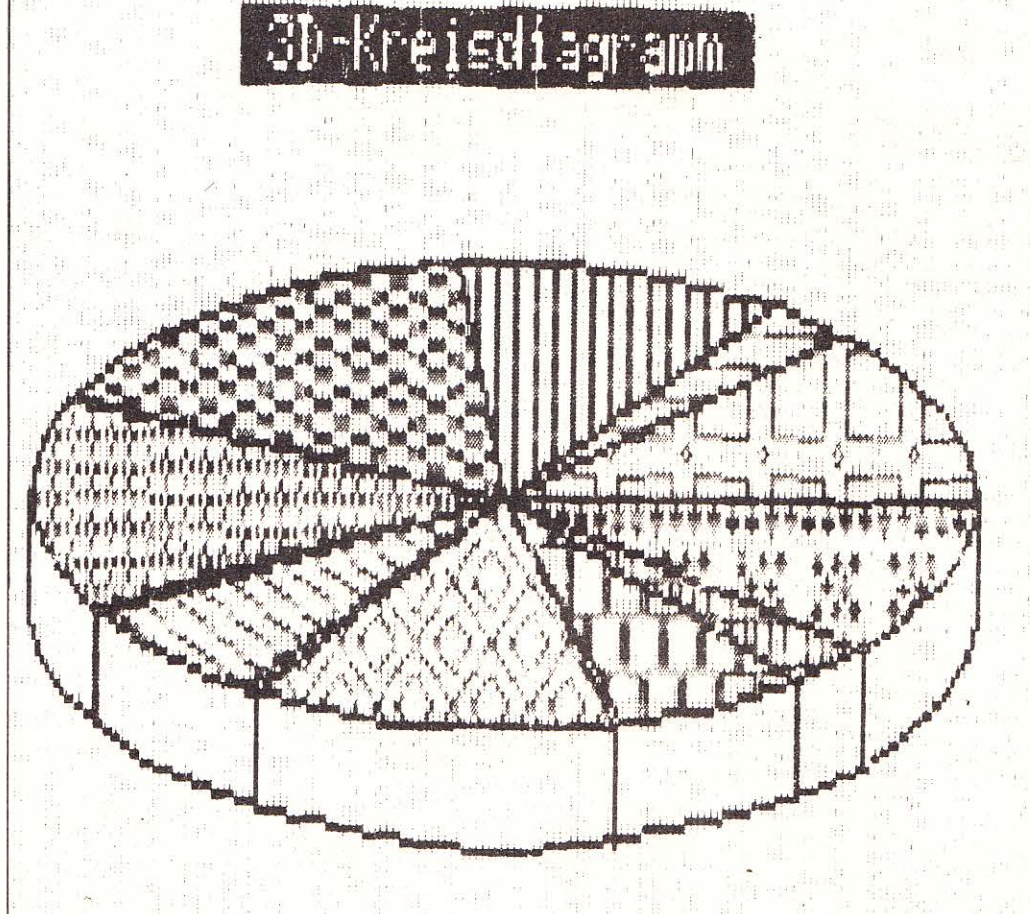
Für Leute, die's bunt mögen, ist der Okimate 20 der richtige Drucker. Auch für den, der schon einen anderen Drucker besitzt, wäre er ein prima Zweitgerät, um mal nachts um 12 ein Listing auszudrucken, ohne daß die Nachbarn rebellisch werden, oder eben um eine farbige Hardcopy vom Bildschirm zu erstellen.











Der Okimate 20 macht – wie die Fotos auf dieser Seite zeigen, hervorragende Farbdrucke vom Bildschirm. Aber er weist auch einen kleinen mechanischen Fehler auf: in regelmäßigen Abständen ist „Linefeed" offensichtlich unkorrekt.

DER KLEINE SPEICHER-RIESE: 800 KB AUF EINER FLOPPY!

Commodore hatte wieder "eine gute Idee": Angesporni vom Erfolg des Amiga und dessen Speichermöglichkeiten auf 3.5-Zoll-Disketten (die kleineren "Commodores" von C 16 bis 128 PC benützen in der Regel die 5.25-Zoll-Floppydisks), wurde ein neues Laufwerk auf den Markt gebracht, das eben auf diese Größe eingestellt ist: die Diskettenstation 1581.

Geradezu wie ein "Zwerg" gegenüber seinen großen Brüdern 1541, 1570 und 1571 mutet diese 1,5 Kilo schwere und 14 x 23 cm große Floppystation an, vor allen Dingen, wenn man sie nebeneinander sieht. Aber, so wie im richtigen Leben sollten Sie die Kleinen nicht unterschätzen.

Zunächst wurde der "Winzling" am seriellen Eingang unseres 128 D angeschlossen und das Netzteil (ja, Sie haben richtig gelesen, diese Floppy braucht so eins!) mit ihr verbunden. Wenn wir uns allerdings die Ausmaße ins Gedächtnis rufen (sie ist auch nur 6.3 cm hoch), ist natürlich klar, daß dieser Transformator nicht mehr ins Gehäuse gepaßt hätte. Das Laufwerk ist mit einem 6502A-Mikroprozessor, 32 Kbytes ROM und 8 Kbytes RAM

BENUTZT SELBEN DISKETTEN-BEFEHLE

ausgestattet. Im Lieferumfang enthalten ist die bekannte "Test/Demo-Diskette", die ähnlich wie jene für die 1571er nützliche Files und Routinen für die Arbeit mit dieser Floppy enthält, (doch davon später) sowie das obligatorische Handbuch, unserem Testgerät lag allerdings nur eines in Englisch bei. Wem das nicht so liegt, kann aber jederzeit anhand des beigefügten Gutscheins so eine Anleitung in Deutsch bei Commodore kostenlos anfordern.

Unsere erste Tätigkeit: das Formatieren einer Arbeitsdiskette. Hier werden in keiner Weise neue Befehle oder eine andere Schreibweise verlangt, als Sie es von Ihrer bisherigen Floppystation oder Ihrem Computer gewohnt sind:
OPEN 1, 8, 15, "NO: TESTDISK, 01": CLOSE 1 für den C 64 und
HEADER "TESTDISK", D0, I01 für den 128er. Erfreulich fällt im Gegensatz zur Floppy 1541 beispielsweise auf, daß dieser Vorgang nahzu geräuschlos von sich geht. In Kauf nehmen müssen Sie allerdings, daß sich an der Zeitspanne für's Formatieren nichts ändert: unser Testgerät benötigte exakt 190 Sekunden. (Wie wir's von der 1541er gewohnt sind!).



Trotz kompakterer Ausmaße "packt" die 1581 mehr Bytes auf eine Diskette als die 1571

3160 BLOCKS FREI!

Erstaunt werden Sie aber sein, wenn Sie das "Directory" mit dem üblichen Befehl einlesen: 3160 freie Blocks stehen auf dieser Mini-Diskette zu Ihrer Verfügung! Das entspricht umgerechnet 808.960 Bytes oder etwa 4 Diskettenseiten der herkömmlichen 5.25-Zoll-Disks. Zudem werden 40 Spuren statt 35 angelegt. Durch den Hardware-Aufbau solcher Disketten ist es allerdings mit der "Unart" vieler C 64- oder 128-PC-Anwender vorbei, sich eine weitere Schreibschutzkerbe einzulochen und die Disk doppelseitig zu

verwenden, nun bei diesem immensen Vorrat an Speicherplatz wohl unnötig. Wie Sie es gewohnt sind, wird auch hier ein "Directory", ein Inhaltsverzeichnis, auf die Disk geschrieben, das wie bekannt aufgerufen werden kann: LOAD"$",8 oder beim C 128 per Druck auf die F3-Taste. Falls Sie einen C 64 oder einen 128er als Einzelgerät besitzen, ergeben sich aus dem Anschluß der Floppy 1581 keine Probleme, denn Sie können als Geräteadresse dafür die "8" wählen, bei einem 128 D ist diese Nummer allerdings schon an das interne Laufwerk 1571 vergeben.

GERÄTEADRESSE PER DIP-SCHALTER EINSTELLEN

Dann müssen Sie an der Gehäuserückseite der 1581er die entsprechenden DIP-Schalter zur Einstellung dieser Geräteadresse benutzen (ist im Handbuch ausführlich erläutert, außerdem sieht die Floppy 1571 ebenfalls so eine Einstellmöglichkeit vor, was den Usern derselben hinlänglich bekannt sein dürfte.)

Bei der 1541 oder beim eingebauten Laufwerk des 128 D ist das nur mit einem kleinen Software-Programm bzw. einem "operativen" Eingriff möglich, diesem eine neue Geräte-Nummer zuzuweisen.

Bis zu 296 Files, je nach Speicherplatzverbrauch, lassen sich auf so einer formatierten 3,5-Zoll-Diskette "saven". Und da sind wir auch schon beim Thema: Wie retten Sie Ihre C 64- oder C 128-Software, die noch auf den "alten" 5.25-Zoll-Scheiben existiert, auf die "neuen", kleineren ? Tja, da muß man sich

KOPIEREN VON 5.25-ZOLL AUF 3.5-ZOLL

schon hinter dem Ohr kratzen, denn die üblichen Diskkopierprogramme (in der Mehrzahl für die 1541 entwickelt) funktionieren eben nur mit einem Laufwerk, und da dürfte es schon ein bißchen schwierig sein, eine 5.25-Zoll-Disk in einem 3.5-Zoll-Schacht unterbringen oder umgekehrt, abgesehen davon, daß solche Kopierprogramme ganz erheblich ins DOS (Betriebssystem) der Floppy 1541 eingreifen und mit einem etwas anders konzipierten DOS wie das der 1581 nicht zusammenarbeiten. Auch dieses Problem wird 1571-Benutzern hinlänglich bekannt sein. Am besten geht's mit einem Diskettenkopierprogramm für 2 Laufwerke, wobei die neue 1581er 28 das Laufwerk 2 mit der Geräteadresse "9" verkörpert. (Bitte sehen Sie in Ihrer Software-Sammlung nach, ob Sie so eine besitzen und probieren Sie es aus!)

Da der einfachste Weg in der Regel auch der beste ist, sollten Sie sich die Zeit nehmen, über die Floppy 1541 (oder 1571) ein Programm einlesen zu lassen, sprich in den Computer zu laden und anschließend mit dem SAVE-Befehl auf die "Neue" zu speichern. Mühevoll und zeitraubend, aber datensicher. Scheitern wird man aber spätestens dann, wenn noch zusätzliche binäre Files auf der Quelldiskette vorhanden sind, z.B. Grafikbilder, Maschinenprogramme o.ä., die das Hauptprogramm zum Nachladen braucht. Da müßten Sie zunächst – nach dem Laden mit "< Name >", 8,1" die Anfangs- und Endadresse auslesen, dann dieses File "absolute", d.h. unter Verwendung dieser Adresse "SAVEn" usw.



Die Rückseite mit seriellen Anschlüssen und Dip-Schaltern ähnelt der 1571

Die meisten sogenannten "Fastback-Ups" oder "Speed-File-Copies" waren zum Scheitern verurteilt. Kopierprogramme, die für 2 Laufwerke vorgesehen sind, machen aber keine Schwierigkeiten, sofern Sie die Geräteadresse für die 1581er auf "9" ändern. Angenommen, Sie hätten nun Ihre Programme und Daten auf den neuen 3.5-Zoll-Disketten untergebracht, und wollen nun eben eine solche

HILFREICHE PROGRAMME AUF DER TEST-/DEMO-DISK

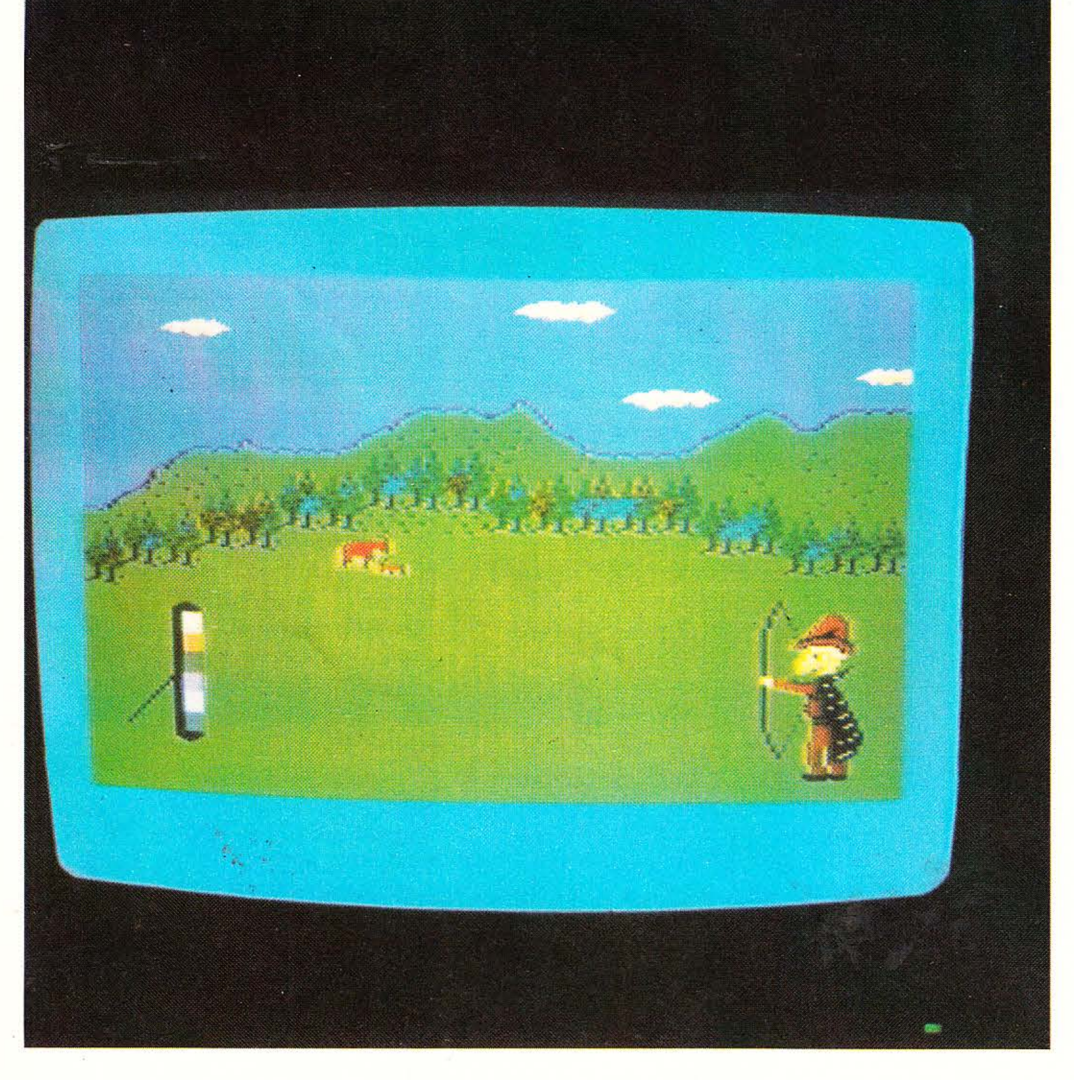
vervielfältigen, bzw. einige Files daraus kopieren. Dafür gibt's bereits Software. Laden Sie Ihre Test/Demo-Diskette; außer vielen kleinen Tips & Tricks, hilfreichen Programmen wie z.B. "Unscratch", finden Sie darauf auch ein Kopierprogramm für einzelne Files oder für eine ganze Diskette, wohlgemerkt: jetzt in 3.5-Zoll-Größe und mit der Floppy 1581. Die Kopierzeiten können zwar für eine ganze Disk fast 20 Minuten betragen, da es sich um Basic-Kopierprogramme handelt, aber findige Maschinenprogrammierer werden sicher bald auch für diese Diskettenstation Kopier-Tools entwickelt haben, sofern – ja, sofern sie vom Käufer angenommen wird.

FAZIT:

Die Vorteile, auf die wir vorher eingegangen sind, liegen klar auf der Hand: geringe Maße, daher platzsparend, zumindest doppelt so großer Speicherplatz wie auf herkömmlichen 5.25-Zoll-Disketten, kompatibel zu allen Homecomputern von Commodore und nicht zuletzt kein Erlernen neuer Befehlssequenzen, um die Floppystation zu betreiben. Der Händlerverkaufspreis soll sich irgendwo unter DM 600,– bewegen (wir glauben aber, daß hier das letzte Wort noch nicht gesprochen ist!). Aus den bereits genannten Gründen ist jedem Neuanfänger, der sich einen Homecomputer von Commodore zulegen möchte, zum Kauf dieser Floppystation zu raten. *(hb)*

KINGSOFT SCHLÄGT WIEDER ZU

Software made in Germany, dieser Wahlspruch des Softwarehauses Kingsoft dürfte der Allgemeinheit inzwischen bekannt sein. Daß hinter diesem Wahlspruch durchweg erstklassige Programmierer stehen, ist ebenfalls bekannt. Nun schickten sich die Kingsoftler erneut an, dies unter Beweis zu stellen und stellten uns ihre neuesten Erzeugnisse rund um den C 16 vor. Soviel vorweggenommen: Nicht nur erstklassige Spiele, sondern auch hervorragende Arbeitswerkzeuge waren dabei.



QUIWI

Dieses Spiel dürfte den meisten bereits bekannt sein. Kingsoft vertreibt das ganze seit neuestem auch für den C 16. Der Spielinhalt ist schnell erklärt, was allerdings nichts über die Qualität aussagt. Quiwi ist ein Ratespiel, das bis zu zwölf(!) Personen spielen können. Einsamkeit verursacht das ganze also kaum.
Und nun zum eigentlichen Spiel: Der Käufer erhält eine Diskette, deren beide Seiten jeweils randvoll gepackt sind. Dies gibt schon Auskunft über den Umfang des Spieles. Exakt 3960 Fragen warten auf die Spieler.
Nach dem Laden des Programmes fragt das Programm ab, ob die richtige Diskettenseite eingelegt ist. Dies ist wichtig, wenn Sie alle Fragen einer Seite beantwortet haben oder auswendig kennen. Bei rund 2000 Fragen pro Diskettenseite dürfte dies eine Weile dauern.
Nach dem Einlesen gelangen Sie in das Menü zur Namenseingabe. Wie gesagt, sind hier bis zu zwölf Personen möglich. Nach dieser Prozedur geht das eigentliche Spiel los. Quiwi, der Spielheld, steht auf einer Wiese und spannt den Bogen. Mit der Feuertaste des Joysticks können Sie den Pfeil auf eine laufende Zielscheibe abfeuern, deren Farben verschiedene Wissensgebiete markieren. Insgesamt sind sechs Gebiete vorhanden: Verschiedenes, Erdkunde, Geschichte/Politik, Sport/Humor, Kunst/Literatur und Wissenschaft/Technik. Mit etwas Übung und Geschicklichkeit hat man bald heraus, wie die einzelnen Farben getroffen werden. Ziel des Spieles ist es, in jedem Wissensgebiet die vorgegebene Anzahl von richtigen Antworten zu geben (diese Anzahl läßt sich von eins bis drei bestimmen). Der Spieler gelangt nun in das Menü, in welchem



FRAGEN ÜBER FRAGEN BEI QUIWI

die Punktestände verzeichnet sind. Das betreffende Wissensgebiet wird nachgeladen, der Spieler erhält seine Frage. Nun muß diese Frage richtig beantwortet werden, dann darf man sich die Lösung zeigen lassen. Nun kann man (nicht schummeln!) angeben, ob die eigene Antwort richtig oder falsch war, oder ob keine Wertung erfolgen soll. Keine Wertung gilt zum Beispiel, wenn sich die Mitspieler uneinig sind, ob die Antwort richtig oder falsch war. Wurde die richtige Antwort ge-

geben, erhält der Spieler einen Punkt unter dem betreffenden Wissensgebiet. In dieser Weise geht es nun weiter, bis der Spieler eine Frage falsch beantwortet. Dann ist der nächste Spieler an der Reihe. Wenn nun ein Spieler alle Wissensgebiete mit ausreichend richtigen Antworten durchgearbeitet hat, gelangt er in die Endrunde. Nun dürfen die Mitspieler das Wissensgebiet bestimmen, für das der Endrundenspieler antworten muß. In der Endrunde muß der Spieler noch einmal drei Fragen richtig beantworten. Antwortet er falsch, wird eine richtige Antwort wieder abgezogen.
Quiwi läßt sich wohl am besten mit dem Spiel „Trivial Pursuit“ vergleichen, das im vergangenen Jahr wahre Verkaufsrekorde erreicht hat. Es ist also sehr wahrscheinlich, daß diese Computervariante des Fragenspieles sehr viele begeisterte Käufer finden wird.
Übrigens: Wer tatsächlich alle Fragen durch haben sollte (falls dies überhaupt gelingt), für den bietet Kingsoft noch einen zusätzlichen Service an. Es können in naher Zukunft weitere Datendisketten geordert werden, die wieder voll mit Fragen sein werden.

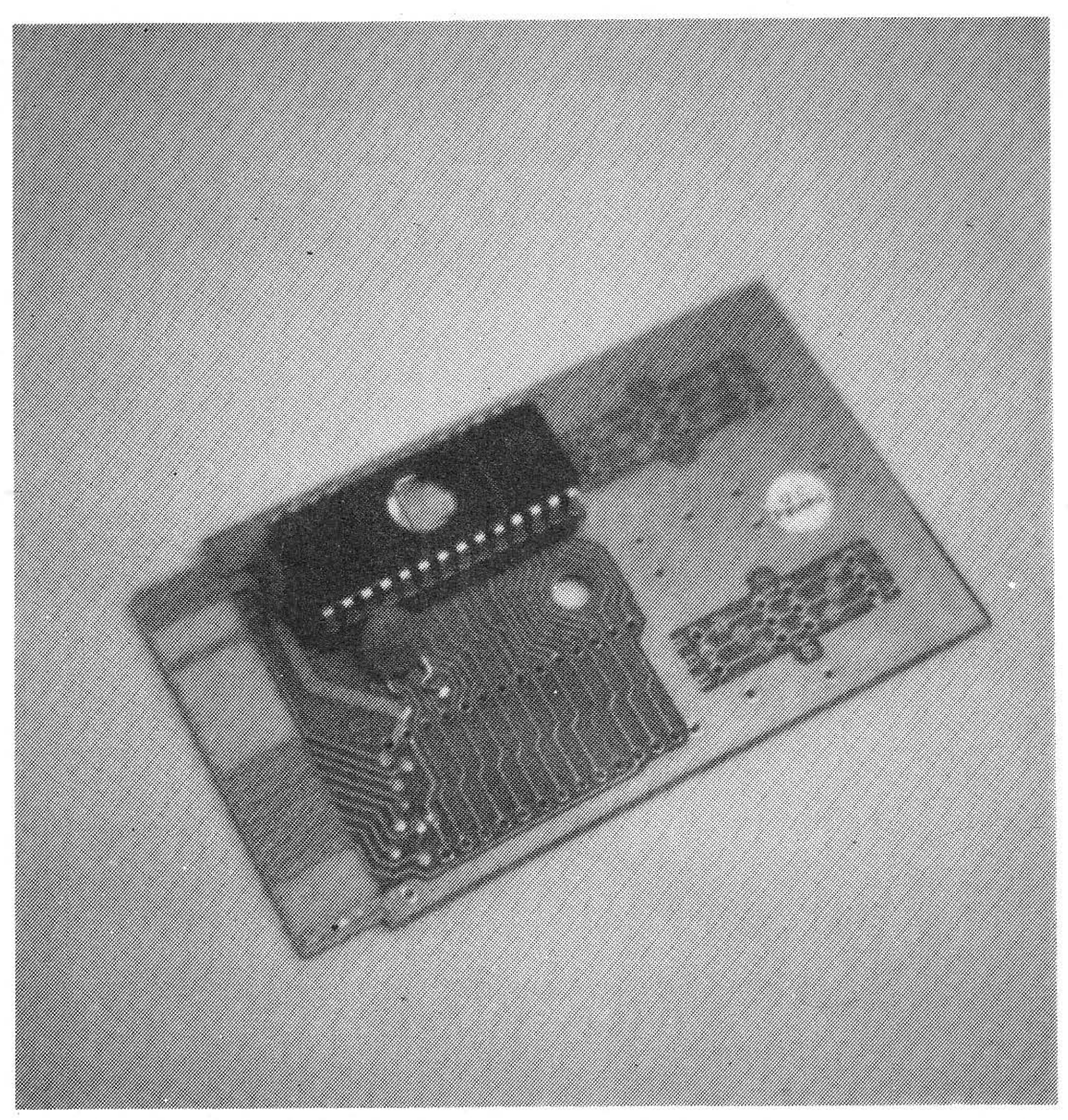
MIKRO DATEI

Nicht jeder möchte mit seinem Rechner nur spielen. Die eigentliche Aufgabe, nämlich die Übergabe von lästigen Routinearbeiten an das Gerät, steht auch bei den Besitzern eines C 16 oder Plus/4 im Vordergrund. Für diesen Zweck bietet Kingsoft unter anderem das Programm „Micro Datei“ für den C 16 an. Dateiverwaltung für den Haushalt, aber auch für professionelle Zwecke offeriert dieses Programm mit etlichen Features.

Das wichtigste Problem einer Dateiverwaltung ist vor allem die Eingabemaske. Darunter versteht man die Felder, unter denen der Benutzer seine Daten eingeben kann, die zu verwalten sind. Dies können sowohl Schallplatten- oder Videoarchive sein, als auch Adressdateien für ein Kleinunternehmen.
Bei Microdatei besteht nun die Möglichkeit (beim ersten Programmstart sogar die Pflicht), seine eigenen „Masken“ zu erstellen. Dies sieht wie folgt aus: Beim Programmstart kann man sich entscheiden, ob die eigentliche Dateiverwaltung oder der Maskengenerator geladen wird. Mit dem Maskengenerator kann nun die gewünschte Eingabemaske nach persönlichem Bedarf erstellt werden. Dies ist so besonders wichtig, weil jeder hinsichtlich seiner Dateien eine bestimmte Vorstellung hat, wie diese genau zu gestalten sind. Mal soll ein Zusatztext möglich sein, mal soll auch die Telefonnummer vermerkt sein. Mikrodatei ermöglicht

INDIVIDUELLE MASKENERSTELLUNG

also, diese Wünsche zu berücksichtigen. Die fertig erstellte Maske kann nun unter einem prägnanten Namen auf die Diskette gespeichert (auch eine Kassettenversion ist erhältlich) und beim Start der Dateiverwaltung eingeladen werden. Durch diese Vorgangsweise hat der User noch einen zusätzlichen Nutzen: Er kann dieses Programm gleichzeitig benutzen, um seine Videokassetten zu archivieren, die Telefonnummer der Geschäftspartner sinnvoll abzulegen oder die Urlaubstermine bis ins Jahr 2000 zu verzeichnen. Hierfür muß nur eine spezielle Maske erstellt werden, die man am besten auf eine separate Diskette speichert. Dann kann mit dieser Diskette als Speicherdisk gearbeitet werden. Die Funktionen des Dateiprogrammes sind durchaus professionell zu nennen. Es fehlt eigentlich an nichts, was im täglichen Gebrauch benötigt wird. Folgende Funktionen stehen im Programm zur Verfügung:
– Directory-Anzeige



- Löschen
- Einfügen
- Korrigieren
- Suchen
- Sortieren

sowie drucken. Natürlich war dies nur eine Auflistung der wichtigsten Befehle, Mikrodatei kann noch mehr. Es ist beispielsweise eine Help-Funktion in das Programm integriert, die mit Aufruf durch die HELP-Taste zu jedem Befehl einen Hilfstext anzeigt. Auch die Bedienungsanleitung ist trotz etwas knappem Umfang informativ und vor allem gut verständlich gehalten. Der Preis von 29 Mark für dieses Programm scheint uns als durchaus angemessen, für die durchschnittlichen Datei-Arbeiten genügt Mikro Datei allemal.

TURBO PLUS

Den absoluten Knüller aus Kingsofts Neuheiten-Kiste haben wir uns bis zum Schluß aufgehoben: Das Modul Turbo Plus. Dieses Modul stellt für die Benutzer des 16ers oder Plus/4 wohl eine der sinnvollsten Erweiterungen dar, die auf dem Markt sind. Es beinhaltet folgende Features:

- Turbo Tape
- Basic-Erweiterung
- Listing-Scroller
- Plus/4-Erweiterung.

Die Punkte im einzelnen:

1. Turbo Tape

Die entnervenden Lade- und Speicherzeiten beim Kassettenbetrieb haben wohl schon manchen zum Verzweifeln gebracht. Durch die Verwendung des Modules kann nun aufgeatmet werden. Ohne schwierige Befehlsketten oder große Aufruf-Routinen kann man den Kassettenrekorder weiterhin nutzen wie gewohnt. Mit den üblichen Kommandos (z.B. LOAD oder SAVE) schaltet sich das Modul automatisch dazwischen und steigert die Geschwindigkeit bei Lese- und Schreibvorgängen um ca. das Zehnfache.
Weiterhin bleibt während des Ladens der Bildschirm eingeschaltet, so daß man den Ablauf verfolgen kann. Sollten Programme geladen werden, die ohne Turbo-Tape gespeichert wurden, kann man das Modul trotzdem im Expansionsport belassen. Es kann entweder durch Zusatz einer Sekundäradresse beim Laden (z.B. LOAD "NAME",1,2) signalisiert werden, daß Turbo-Tape nicht benutzt werden soll. Oder Sie schalten mittels dem Befehl TUROFF das Tubo-Tape völlig aus und nutzen die Kassette, als ob das Modul nicht vorhanden wäre. Natürlich kann dieser Befehl durch TURON wieder rückgängig gemacht werden, das Turbo-Tape schaltet sich wieder ein.

2. Basic-Erweiterung

Kaum jemand würde das Basic 3.5 der C 16 oder Plus/4 als mager bezeichnen, vergleicht man es mit dem des 64ers. Trotzdem fügt das Modul noch insgesamt 14 Befehle hinzu, deren Leistung gar nicht genug zu loben ist. Eine kleine Liste soll hier die wichtigsten der neuen Basicbefehle erläutern.
- CHANGE. Dieser Befehl ermöglicht dem Programmierer, ähnlich wie bei einer Textverarbeitung, das gesamte Programm oder auch nur Teilbereiche nach bestimmten Befehlen oder Variablen zu durchsuchen und diese durch andere zu ersetzen.
So können Variablen ohne lange und mühselige Arbeit umbenannt werden, Basic-Befehle durch sinnvollere ersetzt werden und so weiter.
- DUMP. Mit diesem Befehl listet das Turbo Plus sämtliche Variablen des Programmes, welches gerade im Speicher ist. Dies ist vor allem bei längeren Programmen äußerst praktisch, um nicht die Übersicht zu verlieren.
- DMERGE. Mit diesem Befehl können verschiedene Programme von Diskette aneinandergefügt werden. Diesen Vorteil wissen Disketten-Benutzer wohl ausreichend zu schätzen.
- MERGE. Funktioniert analog zu DMERGE, aller-

dings können Sie hierbei auch Programme von Kassette aneinanderfügen.

– FIND. Eine furchtbar praktische Erweiterung. Dieser Befehl durchsucht ein Programm nach angegebenen Zeichenfolgen und listet alle Zeilen auf, in denen diese verwendet werden.

Gerade für die Besitzer eines unerweiterten 16ers sei hier ein Einsatzbeispiel gebracht, welches den Nutzen dieses Befehles verdeutlicht: Sie haben ein ellenlanges Programm, das den gesamten Speicher füllt und bei Start deswegen zusammenbricht. Nun wollen Sie sämtliche REM-Zeilen löschen, um kostbaren Speicherplatz zu sparen. Nichts einfacher als das: Sie geben ein FIND REM, daraufhin wird das gesamte Programm nach REM-Zeilen durchsucht. Diese werden aufgelistet. Nun schreiben Sie sich die betreffenden Zeilennummern auf. Der nächste Schritt besteht darin, herauszufinden, ob diese

MIT DEM FIND-BEFEHL AUF DER JAGD

REM-Zeilen Sprungziele sind, ob sie also innerhalb des Programmes angesprungen werden und eine Löschung der Zeilen zum Programmabsturz führen würde. Nun nehmen Sie den Zettel, auf welchem Sie die Zeilennummern der REM-Zeilen vermerkt haben. Nächster Schritt: FIND (Zeilennummer). Nun durchwühlt das Modul das gesamte Programm nach Zeilen, in denen die Zahlenfolge (beispielsweise 1250) vorkommt und gibt diese auf dem Bildschirm aus. So können Sie feststellen, ob die Zeile 1250 angesprungen wird, wenn nein, kann diese gelöscht werden; wenn ja, können Sie direkt das Sprungziel ändern.

– OLD. Mit diesem Befehl läßt sich ein Programm, welches leichtsinnigerweise durch den Befehl NEW aus dem Basic-Speicher geworfen wurde, restaurieren. Sämtliche Basic-Zeiger und die Programmverkettung werden neu errechnet!

– KILL. Dieser Befehl schaltet die Basic-Befehle des Turbo-Modules aus. Dies waren, wie gesagt, die wichtigsten neuen Befehle, weitere würden den Rahmen dieses Berichtes doch sprengen.

3. Der Scroller

Dieser Scroller ermöglicht folgendes: Mittels der Cursortasten können Sie das aktuelle Basic-Programm nach Belieben nach oben oder unten scrollen. Im Klartext: Statt des umständlichen LIST xxx–xxx können Sie nun schön langsam oder auch sehr schnell aufwärts oder abwärts listen, mittendrin aufhören und eine Zeile abändern und wieder weiterlisten, all dies ohne den geringsten Befehl. Und sollte dieser Scroller einmal nicht benötigt werden, läßt er sich mit SCROLLOFF einfach abschalten.

4. Plus4-Erweiterung

Im Plus/4 befinden sich bekanntermaßen vier eingebaute Programme. Diese aber arbeiten normalerweise nur in Verbindung mit einem Diskettenlaufwerk, sehr zum Ärger derjenigen, die nur ein Kassettengerät besitzen. Durch Turbo Plus ist dieses Manko nun aufgehoben, drei der vier Programme können nun auch mit Kassette betrieben werden (Text, Grafik und Kalkulation). Die Dateiverwaltung kann leider nicht mit Kassette betrieben werden, da eine relative Datei nur auf Diskette möglich ist. Doch damit nicht genug: Das Turbo Plus korrigiert auch noch den Fehler, der in bestimmten Versionen des Plus/4 auftritt: In der Version V052084 (diese Nummer erscheint nach dem Aufruf der internen Software) traten bei Nutzung der Dateiverwaltung Fehler auf, wenn mehr als 255 Einträge vorhanden waren. Turbo Plus testet nun automatisch, ob dieser Fehler vorliegt und korrigiert diesen.

Dies alles dürfte hinlänglich belegen, was für ein „Teufelsmodul" Kingsoft da auf die Beine gestellt hat. Turbo Plus ist für jeden ernsthaften Anwender des C 16 oder Plus/4 ein absolutes Muß. 49 Mark als Verkaufspreis sind, angesichts dieser Erweiterung, sehr angemessen. Wer öfters selbst programmiert, kann durch dieses Modul einen gewaltigen Zeitvorteil erlangen. Wen wundert's da noch, daß das Testmodul, welches zum Testzeitpunkt das einzige Exemplar von Kingsoft war, die Redaktionsräume nicht mehr verließ? Nach kurzer Rücksprache mit Fritz Schäfer, Geschäftsführer von Kingsoft, durften wir das Vorserienmodell behalten.

PAPER BOY

Kingsoft hat für den Software-Giganten Elite die Umarbeitung dieses Arcade-Games übernommen. Vielen dürfte dieses Spiel beispielsweise vom 64er bekannt sein, jetzt ist jedenfalls die 16er-Version fertig. Das Thema ist bekannt, es geht um das allmorgendliche Zeitungsaustragen mit dem Fahrrad. Es ist erstaunlich, daß trotz des Speichermankos des C 16 eine sehr passable Umsetzung der Grafik auf diesen Rechner geschehen ist. Es kommt der 64er-Version schon ziemlich nahe. Daß bei dieser Grafikleistung der Sound etwas leiden mußte, ist klar; dies wirkt sich allerdings nicht auf den Spielspaß aus. Der Sinn des Spieles dürfte bekannt sein: Es geht darum, in einer Kleinstadt per Fahrrad Zeitungen auszutragen. Mit dem Joystick lenkt man das Gefährt durch die Straßen, per Feuerknopf wirft der Paperboy eine Zeitung. Nun muß gut gezielt werden, denn jeder Wurf muß sitzen. Der Paperboy hat neun Zeitungen. Wirft er eine an ein Zeitungsrohr, so gibt es Bonuspunkte. Trifft er das offene Fenster oder die Stufe vor der Haustür, gibt es jeweils die maximale Punktzahl. Wird daneben geworfen, geht die Zeitung verloren und es gibt keine Punkte. In unregelmäßigen Abständen tauchen Gefahren für den Radler auf:

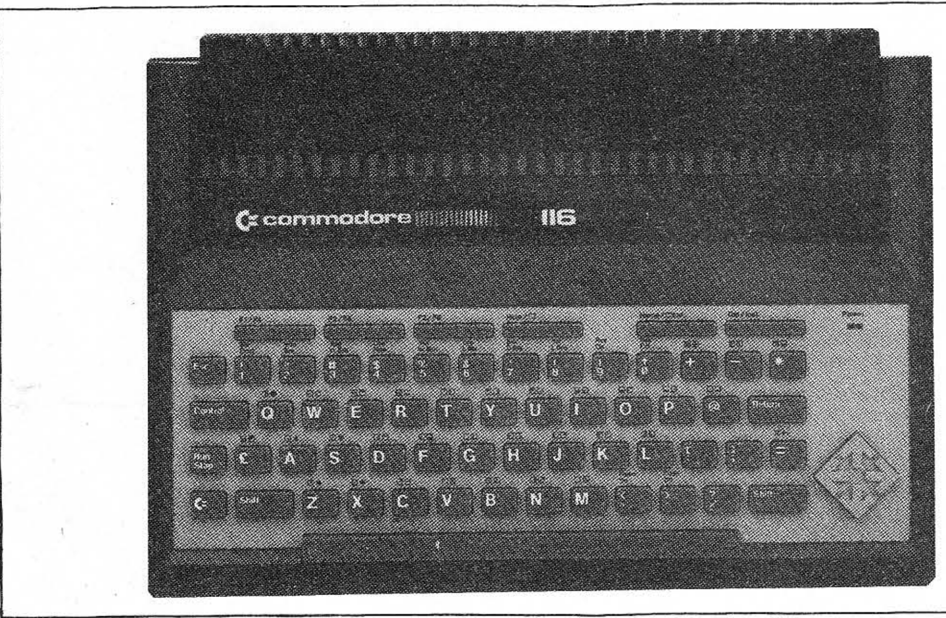
– Auf der Straße sind Gullys vorhanden. Diese zu überfahren, führt zum CRASH.
– Auf den Straßen und Gehwegen liegt allerlei Gerümpel herum. Dieses sollte nach Möglichkeit umfahren werden, um nicht zu stürzen.
– Ab und zu kommen bewegliche Hindernisse, zum Beispiel herrenlose Kinderwagen, einhergefahren. Diesen muß natürlich ausgewichen werden.

Jeder CRASH fordert ein Radler-Leben, drei davon stehen pro Spiel zur Verfügung.

Für einen Preis von 19 Mark kann dieses Spiel den 16er-Besitzern durchaus empfohlen werden, die Umsetzung ist für die beschränkten (Speicher-)Möglichkeiten des 16ers sehr gut geraten.

COMPUTER UND SPIELE „SPIELCOMPUTER" UND ARBEIT

Es gab einmal, ältere Zeitgenossen mögen sich erinnern, selige Zeiten des Computertums. Zu den damaligen Tagen fragte sich der Elektronik-Freak, welchem Gerät er nun den Vorzug geben solle: Der Atari-Spielkonsole oder dem „Volkscomputer" VC 20. In jenen Tagen entschied sich die Zukunft der Homecomputer, die nicht zuletzt auch wegen des Spieleangebotes heranreifte. Was können die drei Einsteigercomputer in dieser Hinsicht bieten?



Es ist wirklich noch nicht all zu lange her, denn anfangs der 80er Jahre wurde der heute schon legendäre VC 20 angeboten. Zu Preisen, bei denen heutigen "Cracks" nur noch das Blut in den Adern gefriert.
Dort auch begann dieser Existenzkampf, der schon im Vorwort beschrieben ist. Denn damals noch mußten sich Spielefreaks entscheiden, ob nun ein reines Telespiel oder ein Heimcomputer die bessere Lösung wären. Die Vergangenheit hat gezeigt, daß die Homecomputer rundweg die beste Veranlagung für alle Zwecke boten.
Darum begannen damals, im Steinzeitalter des Computertums, die ersten Adaptionen erfolgreicher Arcade-Games auf reinrassigen Computern. Wo bei den Spielkonsolen meist knappe hundert Mark pro Spiel berappt werden mußten, konnten die Enthusiasten bereits beste Software für ein Butterbrot erwerben.
Dies war der Beginn . . .

STOFF FÜR MUSEEN

Seitdem sind einige Tage ins Land gegangen. Der VC 20 wird schon eifrig in neudeutschen "Computermuseen" ausgestellt, ehemals stellte dieser Computer einen neuen Standard dar.
Spritedarstellung, vielfältige Bildschirmfarben sowie programmierter "Sound" machten aus den grauen und unmäßig teuren Bürorechnern Spielgeräte für den Heimgebrauch. Der Nachfolger des VC 20, C 64 (Legende schon zu Lebzeiten) räumte und räumt gewaltig ab, wenn es um Spiele geht. Nur allzu gerne weisen die Besitzer dieser Computer (zu recht) auf die wirklich umwerfenden Möglichkeiten betreffend Grafik und Sound auf dem C 64 hin. Mit einem großen Problem. Diese Programmierung gestaltet sich als unglaubliche Schinderei, denn komfortabel war weder das VC 20-Basic noch das des 64ers. Wer es beherrschte (natürlich auch die Maschinensprache), holte aus diesen Geräten Unglaubliches heraus. Nur: Wer unter den Feld-, Wald- und Wiesenprogrammierern hat schon viel Freude daran, endlose Bildschirm-POKEs auswendig zu lernen, die Speicheradressen des Computers besser zu kennen als die eigene Telefonnummer . . .
Und nun gibt es da die 3.5er-Rechner. Diese räumen kräftig mit dem alten Vorurteil auf, keiner könnte es besser als C 64.
Denn was Grafik und Sound betrifft, braucht sogar der Eumel 116 sich kaum hinter den Fähigkeiten des C 64 zu verstecken. Alles ist viel einfacher zu lösen. Beginnend mit dem eingebauten Maschinensprachemonitor, endend mit dem (für Commodore) revolutionären Basic 3.5.
Einziger Nachteil an dieser ganzen Geschichte: Die Softwaremenge. Während es für Commodores Millionenrenner C 64 schon fast so viele Spiele gibt wie verkaufte Geräte vorhanden sind, sind C 16/116 sowie der Plus/4 noch im Hintertreffen.
Doch in England, wo eigentlich der Ursprung dieser Gerätereihe liegt, kann der Spiele-Freak schwelgen. Schwelgen im Softwareangebot, das mittlerweile fast sämtliche erhältliche Spiele des C 64 auch auf den

ALLES, WAS DAS HERZ BEGEHRT . . .

„kleinen" Commodores ermöglicht. Dort beispielsweise ist der schon fast totgesagte Plus/4 absolutes Muß für jedes Softwarehaus.
Ein weiteres Argument gegen die oft hochnäsige Meinung vieler 64er-Eigner, daß soviel Spiele nie für einen anderen Computer angeboten werden, ist etwas fehlerhaft. Zweifellos ist es richtig, daß solch eine Softwaremenge kaum einzuholen ist. Doch stellt sich die Frage, welche 64er-Spiele, die unter den Besitzern rege getauscht werden, überhaupt noch relevant sind. Denn sehr oft sind unter den großspurig offenbarten "über hundert Spielen" wahre Oldtimer, die heutzutage nur noch müdes Lächeln verursachen. Geht man von einer wirklich vergleichbaren Situation aus, dem aktuellen Software-Angebot, sieht die ganze Sache schon sehr viel

besser für die 16er aus. Denn mittlerweile wird so gut wie jedes 64er-Spiel, welches neu auf den Markt kommt oder noch angeboten wird, auch als 16er-Version geliefert. Handelt es sich hierbei um umfangreichere Programme, so gibt es immer noch die Plus-4-Version (z.B. Mercenary, Test in diesem Heft), die auch auf ordnungsgemäß erweiterten 16ern läuft.
Stichwort Raubkopien. Dies stellt ja gerade unter Spiele-Freaks einen nicht zu unterschätzenden Kauf-Anreiz dar. Denn warum keinen 64er kaufen, wo doch alle Schulfreunde schon einen haben? Mit einer unübersehbaren Menge an Spielen, die ja nur kopiert zu werden brauchen? Doch diese Rechnung ist wiederum einseitig.
Denn die Softwareindustrie versucht natürlich mit allen Mitteln, diese Kopiererei zu unterbinden. Da inzwischen bekannt ist, daß es keinen perfekten Kopierschutz gibt, versuchen die Spiele-Hersteller nun eine neue Variante: Es werden immer umfangreichere und komplexere Spiele angeboten, die man ohne Bedienungsanleitung gar nicht vollständig nutzen kann. Und wer hat nun für sämtliche unter der Hand kopierten Spiele auch gleich die Anleitung mitkopiert?



Das Spitzenmodell der 3.5er-Reihe, der Plus 4, ermöglicht sowohl vom Speicherplatz als auch von der Bedienung komfortables Arbeiten.

UNGEWÖHNLICHE METHODEN GEGEN RAUBKOPIEN

Als all zu gutes Beispiel hierfür seien nur die Spiele "Mercenary" oder "Zork" angeführt, die ohne gute Anleitung noch nicht einmal die Hälfte wert sind. Zu alle dem bringen immer mehr Hersteller geradezu üppige Pakete heraus, die zu 50% schon nichts mehr mit dem Computer zu tun haben. So gibt es vom Adventure-Riesen Infocom seit neuestem ein Spiel mit beigefügtem 3D-Comic (samt dazugehöriger Brille, versteht sich), einer "Riech-Karte", welche je nach Umgebung angekratzt wird und so das passende Aroma zum Spielstand vermittelt und und und . . .

ARBEITEN MIT DEN 16ern

So etwas läßt sich nur noch schwer kopieren. Daher sollten die Computer-Aspiranten ihr Augenmerk weniger auf das Vorhandensein von kopierbarer Spiel-Software lenken, als die zukünftige Entwicklung und Verwendung des Gerätes in Betracht zu ziehen.
Weg von den Spielen, hin zur "professionellen" Anwendung. Auch hier gibt es bereits sehr gute Angebote, um mit den 3.5ern auch arbeiten zu können. Dabei ist ganz klar, daß sich kaum jemand so ein Gerät ins Büro stellen dürfte, für die häuslichen Anwendungen genügt die angebotene Software allemal.
Herausragend ist hierbei die Modul-Software zu nennen, die direkt von Commodore sowie über den Fachhandel angeboten wird. Die Rede ist von Script/Plus und Calc/Plus, welche beispielsweise beim Ing.-Büro Stechmann für unter vierzig (!) Mark angeboten werden. Diese Module genügen auch fortgeschrittenen Ansprüchen für Textverarbeitung und Kalkulation. So ist bei Script/Plus beispielsweise ohne weiteres die Erzeugung deutscher Umlaute, von Sonderzeichen und -schriften sowie die Darstellung von bis zu 240 Zeichen/Zeile (bei Bildschirm-Scrolling) möglich. Ein weiterer Vorteil der Modul-Software liegt im Speicherplatz. Dieser ist ja bei den 16 K-Versionen besonders kostbar. Wird der Speicher

MODULSOFTWARE IST DIE BESSERE LÖSUNG

durch ein Software-Textprogramm geschmälert, ist dies mit zumindest einem Nachteil verbunden:
1. Der verbleibende Speicherplatz ist für längere Dokumente zu gering.
2. Um dieses Manko wenigstens halbwegs zu beheben, wird auf umfangreichere Bedienungsmöglichkeiten verzichtet, das Programm wird also "abgespeckt".
Mit der Modulsoftware ist dieser Mangel behoben, denn ein Modul benötigt keinen Speicherplatz, dieser bleibt für den Arbeitsvorgang frei. Weiterhin kann so auch größerer Bedienungskomfort erreicht werden, die Software bleibt qualitativ hochwertig. Wie wir schon in unserem Sonderheft 3/86 und verschiedenen Ausgaben der COMMODORE WELT beschrieben haben, sind diese beiden Module die beste Wahl, wenn es ans Arbeiten geht.
Ein weiteres Gebiet erschließt der in Deutschland neue Lightpen für C 16/116 und Plus/4. Dieser erfüllt alle Anforderungen zur Grafikerzeugung und -steuerung direkt am Bildschirm. So entfällt endlich das von anderen Grafikprogrammen bekannte lästige Umherwan-

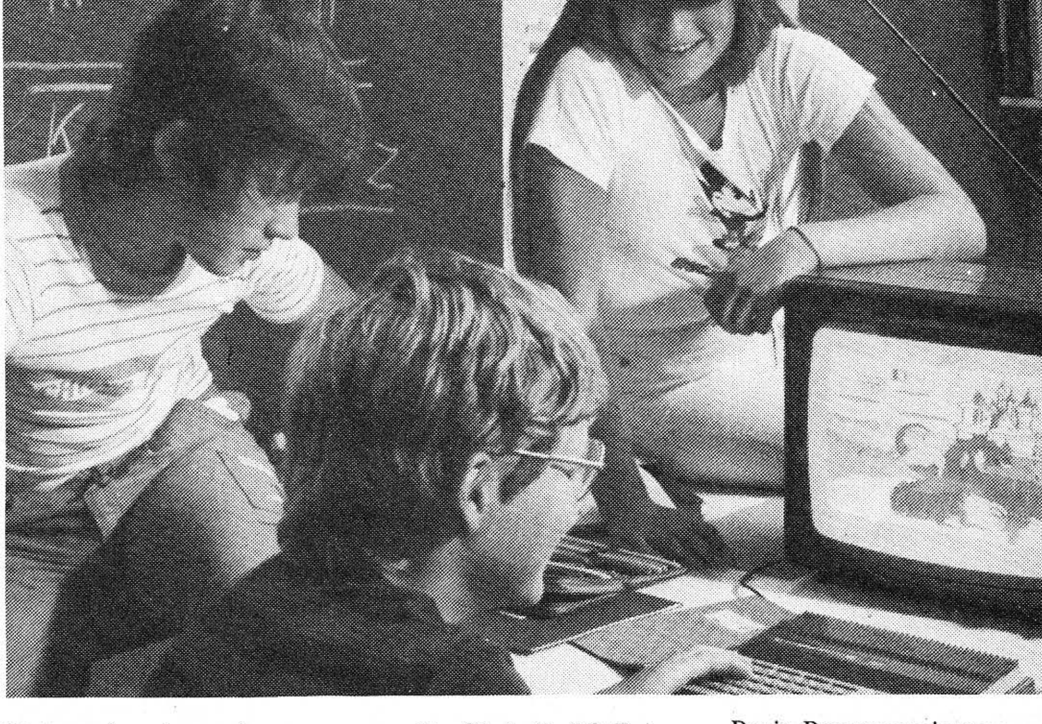
LIGHTPEN AUCH AM C 16!

dern mit dem Joystick. Der Lightpen, welcher zur Zeit nur vom Ing.-Büro Stechmann angeboten wird, ermöglicht unter anderem folgende Funktionen (in Verbindung mit dem dazugehörigen Grafikprogramm):
- Freihand-Zeichnen mit 5 verschiedenen Stiftstärken
- Pinsel-Funktion
- Füllen beliebiger Shapes in 16 verschiedenen Farben
- Verschiedene geometrische Figuren, z.B. Kreise, Vierecke usw.
- Setzfunktion für einzelne Pixels (!)
- Druckroutine für Bildschirmausdruck

und noch viele mehr. Einen ausführlichen Testbericht dieses Produktes, der aus Zeitgründen in dieser Ausgabe leider nicht mehr möglich war, finden Sie in einer der nächsten COMMODORE WELT-Ausgaben.
Der Lichtgriffel wird für 59 Mark bzw. 69 Mark für die Kassetten- oder Diskettenversion erhältlich sein. *T.S.*

16/116/P4: DIE WÖLFE IM SCHAFSPELZ

Die Commodore-Rechner mit dem 3.5er Basic werden oft belächelt und mit Vorurteilen belegt. Doch der erste Schein trügt, denn wer mit diesen Geräten arbeiten will, merkt sehr schnell, was für ein Werkzeug er da in Händen hat. Zugegeben, der 116er ist mit seiner Gummi-Tastatur und dem Hosentaschen-Format nicht gerade als repräsentativ zu bezeichnen. Doch nichtsdestotrotz gehören die Computer C16, 116 und Plus/4 zum Besten, was Commodore je gebaut hat.



Ein mächtiger Vorteil jagt bei diesen Geräten den anderen, alle aufzuzählen, wäre wohl etwas umfangreich. Beginnen wir beim Preis: Dieser liegt mittlerweile für alle drei Computer dieser Baureihe jenseits von gut und böse. Den Gummi-Eumel 116 bekommt man fast schon als Werbepräsent in die Einkaufstüte gelegt, der 16er geht samt Rekorder für knappe hundert Mark über die Ladentische und das Flaggschiff der 3.5er-Reihe, der Plus/4 kostet mit der Super-Floppy 1551 inzwischen weniger als Opa C 64 alleine. Gründe hierfür zu suchen, dürfte mühsam werden. Einer jedenfalls ist wohl direkt bei Commodore selbst zu finden. Mit solchen Preistaktiken will man in diesem Hause offensichtlich das Käuferpublikum an die Marke mit dem C gewöhnen, bei späterem Modellwechsel wird der Computereigner sicherlich einem Rechner des gleichen Herstellers den Vorzug geben.
Auch das Basic ist für solche Computereigner, die nicht nur sinnlos Raumschiffe abballern wollen, nur mit lautem Hurra zu begrüßen. Denn endlich wurde hier eine halbwegs sinnvolle Variante dieser Programmiersprache von Commodore offeriert, wie sie andere Anbieter schon lange im Programm haben.
Da können dann die Besitzer der unzähligen 64er und 20er nur neidisch blinzeln, wenn beispielsweise mit dem Color-Befehl gearbeitet wird. Ohne endlose Poke-Reihen, nur um die Bildschirmfarbe zu ändern. Auch die Grafikbefehle des 3.5er Basic lassen kinderleichtes Programmieren von komplexen Darstellungen zu. Und noch ein Novum wurde hier von Commodore angeboten. Zum ersten Mal sollte mit dem 3.5er Basic ein halbwegs normgerechtes und vor allem kompatibles Basic Einzug in die C-Homecomputer halten. Denn, was noch nicht jeder weiß, Basicprogramme des 128ers laufen auch, natürlich mit kleinen Einschränkungen, auf den Maschinen mit 3.5er Basic und umgekehrt. Betrachtet man Commodores früheres Gebahren in jener Hinsicht, kann dieser Vorgang getrost als ungewöhnlich bezeichnet werden. So sind Einschränkungen nur bei sehr leicht überschaubaren Befehlen wie denen zur Grafik (z.B. 80-Zeichendarstellung auf dem 128er) und Soundsteuerung vorhanden, die auch Computeranfänger mit relativ geringem Aufwand selbst umändern können. Speichermäßig haben die Einsteigercomputer anfangs weniger zu bieten.

LÖTKOLBEN KONTRA SPEICHERMANGEL

Nur der Plus/4 wartet von vorneherein mit 60 freien Kilobytes auf, die anderen beiden Rechner verfügen lediglich über 16 KB. Bis der Lötkolben kommt. Denn wohl kaum ein anderer Homecomputer läßt sich so leicht und vor allem so billig erweitern wie 116 und 16. Eingeläutet wurde diese Tatsache nicht zuletzt durch die COMMODORE WELT, denn unter anderem veröffentlichten wir die erste Umbauanleitung in Deutschland, nach deren Vorbild schon für unter fünfzig Mark neuer Speicherplatz eingeblasen werden konnte. Und zwar satte 60 Kilobyte, die dem allseits beschworenen C 64 zur reinen Basic-Programmierung nicht zur Verfügung stehen. Besonders lustig wird es, wenn diese 60000 Byte in einem 116er werkeln, der wohl eher einem Taschenrechner denn einem Homecomputer ähnelt. So sind dem Eigner eines erweiterten 116ers in seinem Freundeskreis wohl einige erstaunte Blicke sicher, wenn die Einschaltmeldung erscheint.
Auch das Diskettenlaufwerk, welches für den Plus/4 entwickelt wurde, läßt eigentlich keine Wünsche offen. 3 – 4mal

so schnell wie die altbekannte Tante 1541, läuft die 1551 den 64er-Eignern buchstäblich davon. Womit Commodore endlich auch selbst beweist, daß es schneller geht. Der Preis dieser Station läßt überdies wohl kaum Kritik zu, denn billiger sind nur noch PC-Laufwerke.
Außerdem ist diese Station natürlich kompatibel zum Diskettenformat der 1541.
Besonders nützlich ist dieser Geschwindigkeitsvorteil, wenn es um das Arbeiten mit den 16ern und Plus/4 geht. So stehen beispielsweise zur Textverarbeitung endlich akzeptable Zugriffsgeschwindigkeiten zur Verfügung.

GELOBT SEI DER MONITOR!

Ein weiterer riesiger Vorteil dieser Computer ist der bereits implementierte Maschinensprache-Monitor. Hier haben die Tüftler ein wirklich hervorragendes Werkzeug in Händen, um gute Ideen auch angemessen umzusetzen und nicht auf das im Vergleich schwächere Basic zurückgreifen zu müssen. Für diese immense Hilfe müssen Besitzer anderer Systeme immer noch zusätzlich Geld ausgeben, außerdem stellt eine Softwarelösung zur Maschinensprache-Programmierung sicher nicht das Nonplusultra dar.
Programmierbare Funktionstasten – damit warben Commodores Marketing-Strategen bereits für den VC 20. Wohlwollend verschweigend, wie schwer dies zu bewerkstelligen ist. Bei den 3.5er-Modellen nun können besagte Werbetexter wirklich auf die Programmierbarkeit, vor allem die wirklich einfache Programmierbarkeit, hinweisen. Mit der einfachen "KEY"-Anweisung lassen sich umfangreiche Funktionen auf die einzelnen Tasten schreiben. So können unter Verwendung der Abkürzungen für Basic-Befehle sowie entsprechender chr$-Codes komplexe Programmzeilen auf eine einzige Funktionstaste gelegt werden.
Ebenso einfach ist es möglich, die wichtigsten und am häufigsten verwendeten Basic-Befehle direkt auf diese Tasten zu legen. Von Haus aus ist dies ja bereits geschehen, doch mit Sicherheit stellt jeder Anwender andere Ansprüche an die Leistungsmerkmale dieser Arbeitshilfen.
Unser kleiner Tip dazu: Versuchen Sie, welche Tastenbelegung Ihnen selbst am besten gefällt. Schreiben Sie sodann ein Mini-Programm, welches ausschließlich die Funktionstastenbelegung enthält und speichern es entweder auf Kassette oder Diskette ab. Bei jedem Anschalten des Rechners kann diese Belegung nun in Sekundenschnelle eingeladen werden. Sie brauchen lediglich das Programm zu starten und danach zu löschen. Fortan stehen Ihnen die „persönlichen" Funktionen (bis zu einem System-Reset) zur Verfügung. Ein solches „Programm" könnte beispielsweise so aussehen:

```
10 key 1, "directory"
+chr$(13):key 3, "goto":
key 5, "load"+chr$(34)
```

Diesen Einzeiler müssen Sie nur mit RUN starten, danach mit NEW einfach löschen.
Auch die Möglichkeiten der Gerätesteuerung mit den kleinen Geräten sind unglaublich. Wie Sie an anderer Stelle dieses Heftes nachlesen können, sind selbst die umfangreichsten Steuerungen (bei geschickter Programmierung) kein Problem. Ein Redaktionsmitglied benützt seinen privaten Plus/4 zum Beispiel zur Kamera-Steuerung. Hierfür bringt gerade der Plus/4 mit seinem herausgeführten Userport die besten Voraussetzungen mit.
Alles in allem bleibt also die Genugtuung, mit einem Rechner der 3.5er-Reihe ein wirklich sinnvolles Gerät erworben zu haben. Für wenig Geld erhält der Käufer eine Maschine, mit der es sich wirklich arbeiten läßt. Allen Unkenrufen zum Trotz, laut denen diese Computer ja bereits tot geboren und gar nicht marktfähig seien, beweisen sie jeden Tag aufs neue, was in ihnen steckt. *T.S.*

DAS GRAFIKBUCH ZUM C16/116 UND PLUS/4



Obwohl die Commodore Computer der Reihe 16/116/P4 über hervorragende Grafikmöglichkeiten verfügen, hält sich das Handbuch dieser Computer doch bei der Beschreibung der Grafik sehr zurück. Obwohl die Grafikbefehle genau erklärt werden, bleiben doch viele Fragen offen. Diese Lücke deckt nun ein neues Buch aus dem Hause Data-Becker ab. Die Autoren Löffelmann und Plenge versuchen, den C16/116- und Plus/4-Besitzern die Grafikmöglichkeiten ihrer Computer nahe zu bringen. Und dieses, so scheint mir, ist den Autoren auf ca. 230 Seiten auch gelungen. Das Autorenteam erläutert im ersten Kapitel alle Grafikbefehle der Computer ausführlich und mit kleinen Beispielprogrammen und Hardcopys ergänzt, so daß ein Nachschlagen im Handbuch beim Programmieren unterbleiben kann.
Das zweite Kapitel beschäftigt sich mit den Grundlagen der Grafikprogrammierung.
Auch hier wird eine ausreichende Grundlagenbeschreibung gegeben, die sich in diversen Formeln und Diagrammen niederschlägt, die z.B. auch erläutern, warum eine schräge Linie auf dem Bildschirm nun mal wie eine Treppe aussieht.
Das 3. Kapitel ist mit der Überschrift „Anwendungen der Grafikprogrammierung" überschrieben und erläutert die Programmierung verschiedener Programme anhand von Modulen mit ausreichender und genauer Erläuterung der einzelnen Funktionen.
Es gipfelt in einigen Programmen, z.B. einem CAD-Programm, das es ermöglicht, Zeichnungen aller Art auf den Bildschirm zu zaubern. Auch die Freunde der Statistik kommen nicht zu kurz, denn auch für sie gibt es Balken- und Tortendiagramme.
Das nächste Kapitel beschäftigt sich mit den Möglichkeiten einer Hardcopy, der Grafiken auf verschiedenen Druckern, sowie dem Speichern und Laden der Grafiken auf Diskette und Kassette.
Der Anhang enthält einige nützliche Programme, wie Shapeeditor, Datawandler sowie eine Anzahl nützlicher Tips und Hinweise,die man immer wieder einmal benötigt. Auch eine Tabelle, die sämtliche Zeichen und Steuercodes mit ausführlicher Beschreibung enthält, findet man hier.
Alles in allem eine Pflichtlektüre für den angehenden Grafikprogrammierer, der seinen Computer voll ausnutzen will.

Info:
Löffelmann / Plenge
Das Grafikbuch zu
C16/116 und Plus 4
ca. 240 Seiten 29,– DM
ISBN 3-89011-205-6
Data-Becker-Verlag

Drucker-Kunde:

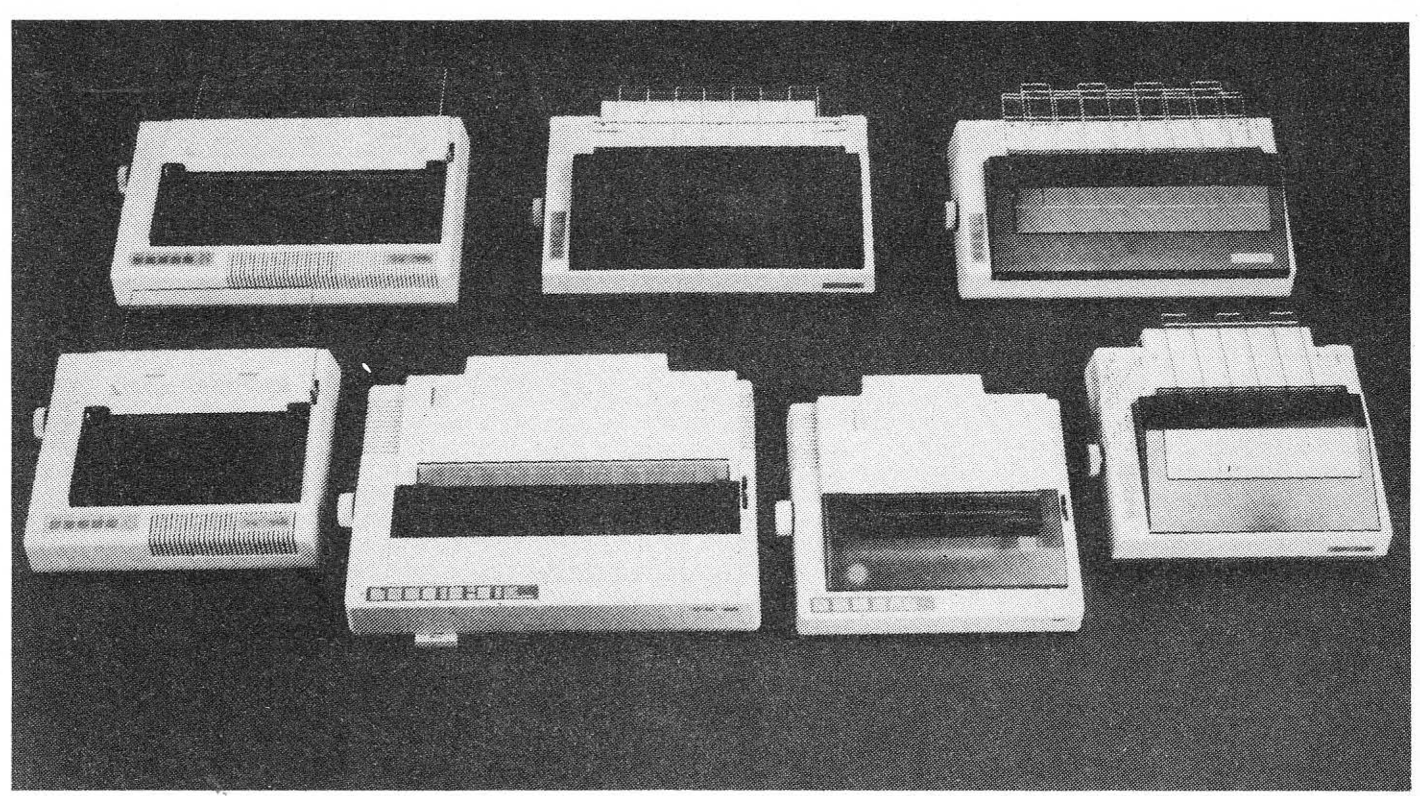
SO FINDEN SIE HOFFENTLICH DEN RICHTIGEN DRUCKER

Neben der Floppy ist der Drucker das wichtigste Werkzeug für einen Computer. COMMODORE SPECIAL sagt Ihnen, was Sie vor dem Kauf eines Druckers beachten sollten.

Drucker gibt es mittlerweile schon wie Sand am Meer. Ob für den Heimgebrauch, zum Ausdruck für kleinere Programme und Listen oder aber für den Profi-Bereich: Für jede Nutzung gibt es mindestens ein halbes Dutzend Anbieter mit verschiedenen Leistungen und verschiedenen Preisen.

„Welche Marke und welchen Typ nehme ich, was ist für mich am Besten?" Diese oder ähnliche Fragen stellen sich jedem Computer-Benutzer, der sich einen Drucker kaufen will.

Selbstverständlich muß diese Frage im Endeffekt jeder einzelne für sich selbst entscheiden. Und gerade im Fall eines Druckerkaufes sollte man sich für diese Entscheidung besonders viel Zeit nehmen.
Das erste, was man tun sollte, ist, in den eigenen Geldbeutel zu schauen, um sich damit klarzuwerden, welche Summe überhaupt ausgegeben werden kann.
Jetzt erst sollte man sich informieren, welche Drucker für welchen Preis angeboten werden.
In der ersten Auswahl fallen alle Typen heraus, die mehr kosten, als Sie bereit oder fähig sind, dafür auszugeben.
In der zweiten Auswahl sollten Sie sich die Frage stellen, ob Ihr Drucker einen bestimmten Standard erfüllen soll.
Gängige Standards sind zum Beispiel Epson- und IBM-Druckformate. Die für den C16 geschriebene Druckersoftware ist entweder für einen Original Commodore Drucker (MPS-Serie) oder aber für einen Drucker mit Epson-Standard bestimmt; ebenso verhält es sich beim Apple IIe.
Haben Sie nun zum Beispiel eben einen C 16 und Sie sind sich sicher, daß Sie keinen IBM oder einen IBM-ähnlichen Rechner in nächster Zukunft kaufen wollen, so brauchen Sie einen Drucker mit Commodore- oder Epson-Standard.
Ansonsten sollte Ihr neuer Drucker sowohl den Epson- als auch den IBM-Standard beherrschen.
Ein weiteres Auswahlkriterium sollte die Grafikfähigkeit des Druckers sein. Ist er grafikfähig, so kann man mit diesem Gerät sowohl normalen Text als auch, mit einem Zusatzprogramm, hochaufgelöste Grafik-



Es gibt sie wie Sand am Meer: Drucker. Hier allein das Angebot eines einzigen Unternehmens: Star.

bilder ausdrucken. (Die meisten modernen Drucker sind grafikfähig.)
Ein weiterer, nicht minder wichtiger Punkt ist die Art der Schnittstelle, die der Printer besitzt.
Schnittstellen gibt es allgemein nur zwei verschiedene: Die serielle und die parallele Schnittstelle.
Die meisten Drucker im Home- und Personalcomputerbereich besitzen parallele Schnittstellen, eine Ausnahme bilden die Commodore.
Will man einen Drucker mit paralleler Schnittstelle an den C16/P4 anschließen, so benötigt man hierfür noch eine besondere Schnittstelle, ein Druckerinterface.
Bei manchen Angeboten ist diese Karte schon dabei, bitte beachten Sie dies!!! Ansonsten sind Mehrausgaben bis zu 250 und mehr Mark möglich!
Hat man diesen wichtigen Punkt auch beachtet, kann man endlich dazu übergehen, den Drucker nach seinen eigenen Leistungen zu testen.
Beginnen wir mit:

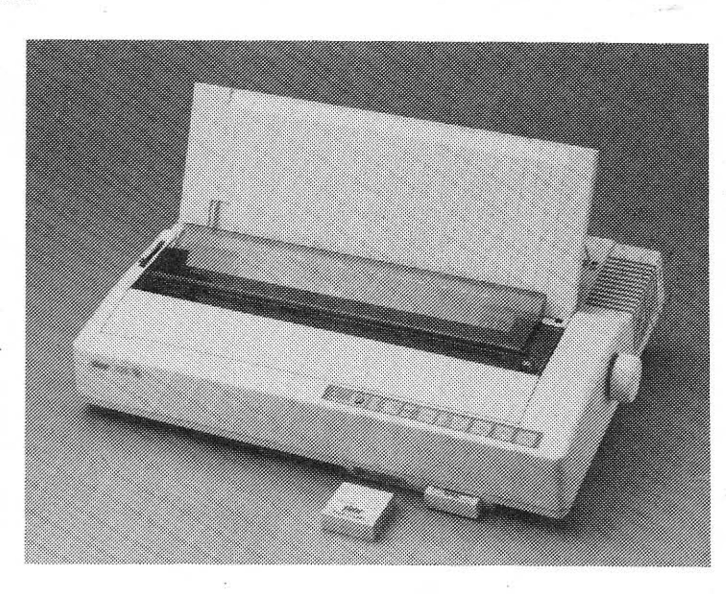
a) Einzelblatteinzug:
Einzelblatteinzug bedeutet nichts anderes, als daß Sie mit diesem Drucker ganz normales Papier (meist bis DIN A4-Format) benutzen können. Ein nicht zu unterschätzender Vorteil.

b) Traktor bzw. Endlosblattverarbeitung:
Dieser Zusatz bedeutet, daß der Drucker Endlospapier mit Lochrand verarbeiten kann.
Auch diese Eigenschaft ist von großer Bedeutung.
Selbstverständlich gibt es viele Drucker, die sowohl den Punkt (a) als auch den Punkt (b) erfüllen; sie sind den anderen Geräten vorzuziehen.

c) Druckgeschwindigkeit:
Die Druckgeschwindigkeit wird in Z/sec. = Zeichen pro Sekunde angegeben. Hat ein Drucker die Angabe 180 Z/sec. so bedeutet dies, daß man von einem Text mit 80 Zeichen pro Zeile jede Sekunde 180 Zeichen abdruckt.
Da die meisten Texte jedoch kürzere Zeilen haben, ist dieser Wert nicht hundertprozentig verbindlich, sondern zu hoch.
Dennoch ist ein Drucker mit einer Geschwindigkeit von 180 Z/sec. um einiges schneller als ein Drucker mit nur 120 Z/sec.

d) NLQ=near letter quality= Schönschriftmodus:
Ein durchschnittlicher, guter Drucker schreibt normalerweise mit einer 8x9 Punkte Matrix.
Schlechtere Drucker, wie zum Beispiel der Commodore MPS 801, haben nur eine 5x7 Punkte Darstellung.
Die Zeichen eines solchen Druckers sind weit schlechter zu lesen, also unsauberer, während die 8x9 Punkte-Drucker weitaus schönere Druckbilder erzeugen.
Achten Sie also vor dem Kauf auf die Druckmatrix Ihres Druckers!!!
Einige Drucker haben nun einen Extra-Modus, NLQ genannt.
Dieser Modus simuliert eine 18x24 Punkte Matrix; der Zeichenausdruck in diesem Modus ist, gegenüber dem Normalmodus, umwerfend; es ist kaum noch ein Unterschied zu einer Schreibmaschine feststellbar.
Dafür ist die Druckgeschwindigkeit auch erheblich verringert, was man aber für besonders wichtige Briefe in Kauf nehmen sollte.
Wichtig ist auf jeden Fall, daß der Drucker die Schönschreibmöglichkeit besitzt.

e) Druckerpuffer:
Ein Druckerpuffer ist ein Speicher, in dem Daten, die vom Computer kommen und die ausgedruckt werden sollen, zwischengespeichert = gepuffert werden.



Moderne Drucker ändern ihre Schriften heute per Einschub-Kassette, Dipschalter umstellen ist nicht mehr „in".

Hat man zum Beispiel ein 3seitiges Listing auszudrucken, so nimmt der Puffer, je nach Größe, alle 3 Seiten auf und man kann, während der Drucker arbeitet, schon am Computer weiterarbeiten.
Bei kleineren Ausdrucken entsteht daher keine ,,Wartezeit", bis der Drucker alles gedruckt hat und sich der Computer erst dann wieder meldet.
Besonders im NLQ-Druck lernt man einen großen Druckerpuffer schätzen, da der Drucker in diesem Modus, wie erwähnt, besonders langsam und die Wartezeit dementsprechend auch länger ist.
Ein vernünftiger Druckerpuffer faßt mindestens 5 KBytes, mehr Speicher ist natürlich besser, aber auch teurer.

f) Zeichenentwicklung möglich:
Einige Druckerfabrikate bieten dem Benutzer noch einen ganz besonderen Leckerbissen.
Auf ihnen kann man eigene Zeichen entwickeln und dann abspeichern.
So wie man auf dem C 16 den Zeichensatz ziemlich leicht ändern kann, so ist dies auch bei solchen Druckern möglich.
Zwar wird diese Eigenschaft fast nie gebraucht, sie ist aber dennoch eine nette und für manche bestimmt nicht unnütze Spielerei.

g) Anzahl der Zeichensätze:
Ein guter Drucker sollte den amerikanischen ASCII-Zeichencode sowie den deutschen Zeichensatz mit Umlauten (äöüß) besitzen.
Viele andere Drucker haben überdies noch französisch, norwegisch, griechisch und andere Zeichencodes fest integriert.

h) Farbband:
Das Farbband stellt für vieldruckende Computerbesitzer eine stetige Ausgabe dar.
Deshalb sollte man sich vor dem Kauf des Druckers nach den Preisen für die Farbbänder erkundigen.
Es ist ebenfalls gut zu wissen, wieviel Zeichen lang ein Farbband hält, d.h. nach wie vielen Druckseiten die Druckintensität nachläßt.
(Manche Farbbänder haben einen Nachfülltank, durch den man man die Farbbandkosten erheblich senken kann. Bedenken Sie also auch dies vor dem Kauf!)

Piloten-Dorado:

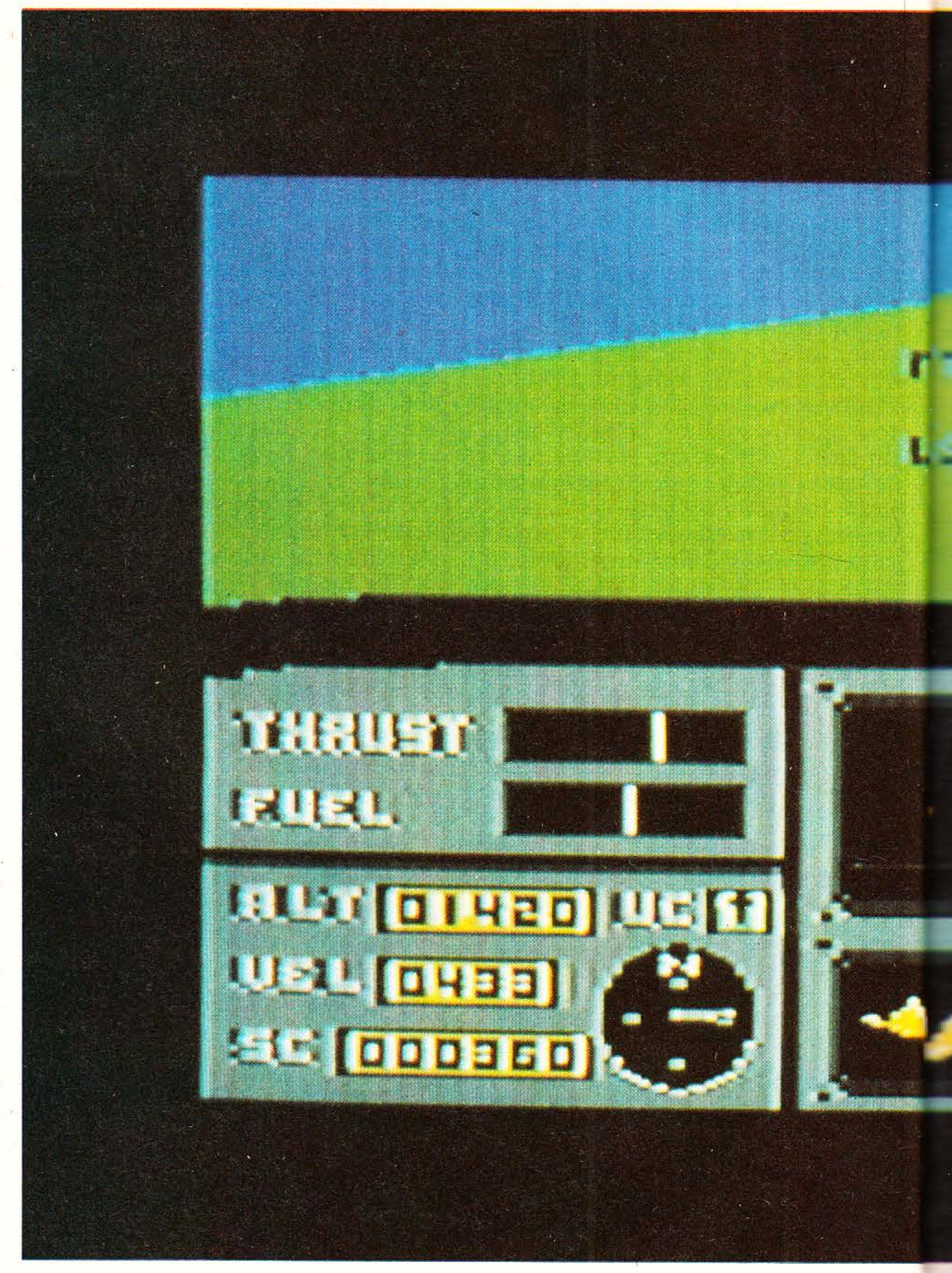
ACE JET FOR FUN!

ACE – Air Combat Simulator, dieser Name läßt den englischsprechenden User schon aufhorchen. Von Cascade Games in England kommt ein Flugsimulator für den C 16, welcher nicht nur das bloße Fliegergefühl vermittelt, sondern auch noch Gelegenheit zu Luftkämpfen gibt. Nun müßen die Plus/4 Besitzer nicht mehr neidisch zu den größeren Rechnern blicken, auch für sie gibt es ab jetzt perfekte Flugsimulation.

Wer ACE auf Diskette erwirbt, wird erst einmal eine Überraschung erleben. Denn: Ein und dasselbe Programm läuft entweder auf dem C 64 oder Plus/4. Wie dies bewerkstelligt wurde, ist Geheimnis von Cascade, es funktioniert jedenfalls. Über die Qualität englischer Spielesoftware für Commodore Computer braucht wohl kaum noch ein Wort verloren zu werden, hier haben sich die Entwickler einmal mehr selbst übertroffen. ACE ist ein vollwertiger Flugsimulator, und es kann noch mehr. Hier wird noch zusätzlich die Möglichkeit gegeben, alle möglichen Kampfsituationen „hautnah" mitzuerleben, nur wird es einem beileibe nicht so einfach gemacht wie bei herkömmlichen Ballerspielen.

PER JOYSTICK UND TASTATUR!

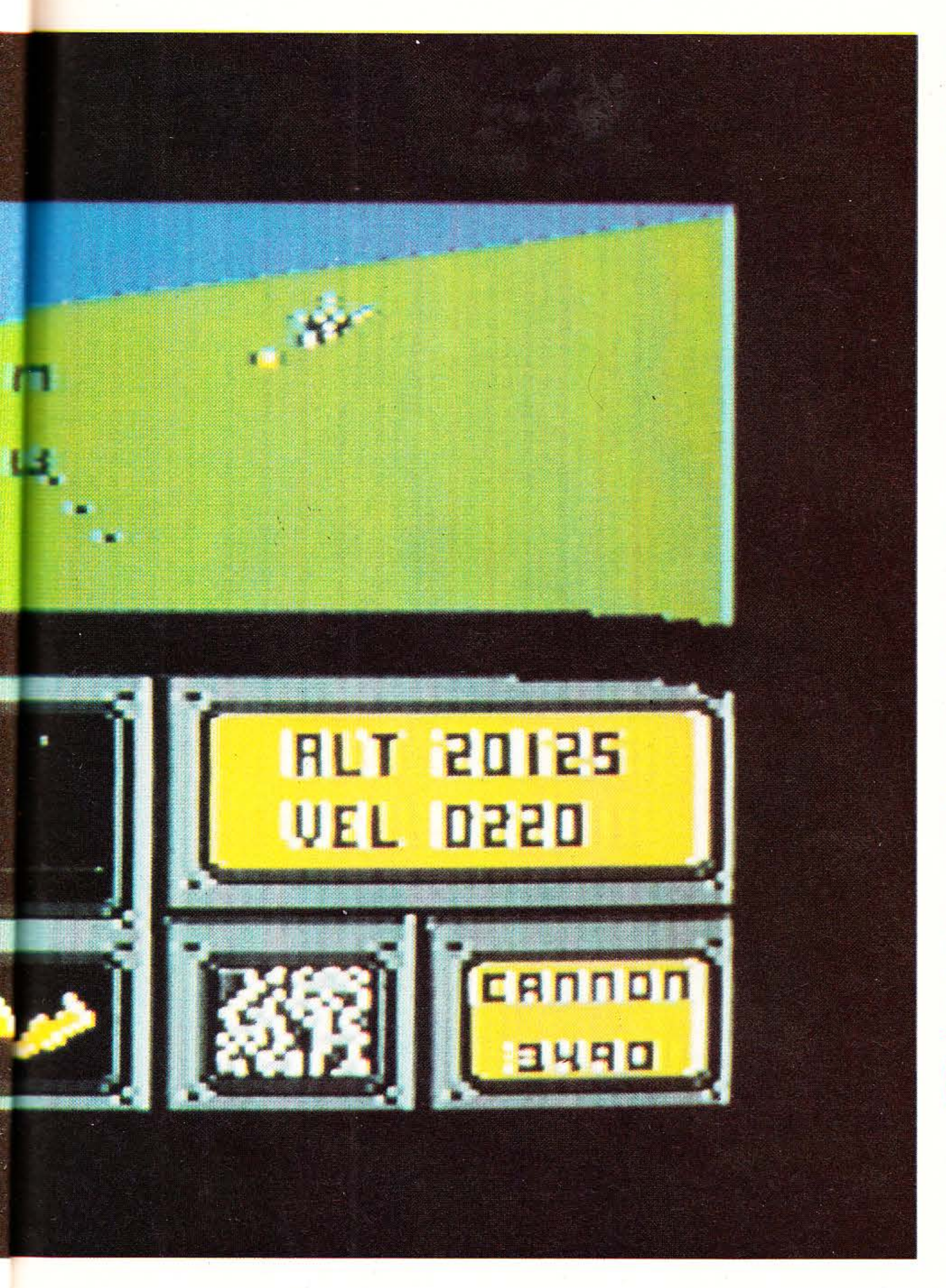
Es gilt vorrangig, die diversesten feindlichen Gerätschaften, von Panzern über Kampfflugzeuge bis hin zu Schlachtschiffen, unschädlich zu machen. Hierzu kann der Spieler zwischen verschiedensten Voreinstelldaten wählen, um die für ihn beste Spielsituation zu erreichen.



Das Bild auf dem Bildschirm: Informationen über nahezu alle Flugzustände.

Es kann z.B. die Flugzeit (Sommer, Winter, Nacht) ausgewählt werden, ebenso kann der Spieler entscheiden, in welcher Art er seinen Einsatz durchführen will, sei es ein reiner Trainingsflug, eine Luft-Luft-Konfrontation oder auch ein Luft-Wasser-Duell usw. Zuerst muß natürlich gestartet werden, was alleine schon für den Anfänger einige Schwierigkeiten mit sich bringen dürfte, sodann kann das Spiel beginnen. Absolut neu ist hierbei, daß die Flugzeugbesatzung auch aus zwei Spielern bestehen kann, welche sich per Tastatur und Joystick die Aufgaben teilen und so noch optimaler agieren können. Es gilt, soviele Feinde wie möglich unschädlich zu machen, was sich diese natürlich nicht ohne weiteres gefallen lassen. So bedarf es mehrerer Spieldurchläufe, um überhaupt zu treffen, ohne auf den sprichwörtlichen Boden der Tatsachen zurückgeholt zu werden.

Zur Steuerung des Flugzeuges steht natürlich primär der Joystick zur Verfügung. Um die vielen Sonderfunktionen zu ermöglichen, muß



jedoch auch die Tastatur in das Spielgeschehen mit einbezogen werden. Diese wird beispielsweise benötigt, um die Bewaffnung auszuwählen (es stehen je nach Einsatz fünf verschiedene Gattungen bereit) oder das Fahrwerk für Starts und Landungen (natürlich beherrscht ACE auch diese) ein- bzw. auszufahren. Wird ACE von zwei Spielern benutzt, entstehen hierdurch zusätzliche Funktionen für den Kopiloten, welcher vor allem für die Abwehr zuständig ist.

Der Pilot bekommt auf dem Bildschirm den Blick aus dem Cockpit dargestellt, hier werden auch sämtliche benötigten Anzeigen aufgeführt. Zur Bewerkstelligung der Aufgaben stehen ein Kompaß, eine Rückenkamera sowie die üblichen navigatorischen Instrumente zur Verfügung. Sollte der Treibstoff knapp werden, was bei zügiger Flugweise relativ schnell der Fall ist, stehen zwei verschiedene Möglichkeiten zum Tanken bereit. In der Luft befindet sich ständig ein Tankflugzeug, welches genauestens auf gleicher Höhe und mit niedriger Geschwindigkeit angeflogen werden muß, um das Vorhaben in die Tat umzusetzen.

Weiterhin ist es möglich, auf einem der drei verschiedenen Heimatflughäfen zwischenzulanden, was zusätzlich den Vorteil der Reparatur und Neubewaffnung bringt.

UMFANGREICHE INFORMATIONEN

Per Tastendruck kann eine Übersichtskarte des Umkreises aufgerufen werden, welche Auskunft über Lage der feindlichen Stellungen, befreundeter Flughäfen oder des Tankflugzeuges gibt. So ist es möglich, auch weiträumige Flugmanöver präzise durchzuführen und sich frühzeitig über angreifende Geschwader zu informieren. Angriffe aus der Luft sind am schwersten zu bewältigen, da die heranfliegenden Maschinen mit allen Tricks und Ausweichmanövern arbeiten, so daß sie nur mit viel Übung vor das Fadenkreuz zu bekommen sind. Sehr hilfreich ist hierbei das eingebaute Radar, auf welchem man die Flugbewegungen verfolgen und entsprechend reagieren kann. Auch die eingebaute Heckkamera leistet hierbei gute Dienste, diese wird allerdings auch als bevorzugtes Ziel der Angreifer gehandelt.

ABSOLUTE PRÄZISION ERFORDERLICH

Wer allzu übermütig ist, kann sich auch fern der Tankstationen wagen und auf See nach Kampfschiffen jagen, die dies natürlich nicht ohne weiteres hinnehmen. Auch hierbei leistet die Übersichtskarte gute Dienste, sie ermöglicht erst das gezielte Ansteuern der Schiffe.

Abschließend bliebe zu sagen, daß ACE wirklich einen ernst zu nehmenden Flugsimulator darstellt, da fast wie beim richtigen Fliegen auf die verschiedensten Einzelheiten geachtet werden muß. Es ist absolute Präzision erforderlich, um dem Ziel ein Stück entgegen zu kommen. Ein sehr gelungenes Spiel, welches sich wohltuend von der Masse der Raumschiff-Ballerspiele abhebt. **Erhältlich ist ACE zur Zeit nur in England bei Cascade Games in Harrogate, aber dem Vernehmen nach soll Kingsoft schon Gespräche führen ...** *T. Seibt*

VIERER-BANDE

Die Spiele für die 16er/P4-Reihe werden immer leistungsfähiger und preiswerter. Wir stellen wieder etliche vor und bewerten sie.

Ein Spiele-Pack für jeden Geschmack: Das Pluspaket II ist ein Vierer-Sparpaket (29 DM) für den C 16 und den P 4, an dem die ganze Familie Freude finden kann: Da gibt's 'Bongo' mit einem heimeligen Häuschen voll Diamanten für die Mutti, zwei Weltraumpilotenabenteuer (Space Pilot und Pilot X) für Vati und Sohn, und Opi schwelgt bei den 'Legionnairen' in seinen Veteranenträumen. Mal ist mehr Geschicklichkeit, mal mehr Vorausplanen gefordert: Tüftler und Fans von schnellen Schieß-Spielen kommen hier gleichermaßen auf ihre Kosten. Das Pluspaket II gibt es als Kassette oder Diskette von Kingsoft, made in Germany.

SPACE-PILOT

Space Pilot tuckert friedlich durch die Galaxien. Da, plötzlich, wird er angegriffen: Bunte, außerirdische Schiffe rasen von allen Seiten auf ihn zu und beschießen ihn mit allerlei Raum-Raketen und ähnlichem gemeinen Zeug. Space Pilot's Raumstation ist zu träge um mit ihr zu fliehen – er hat keine andere Wahl, er muß den Kampf aufnehmen: Schon eine kleine Berührung des Joysticks genügt, um einen wahren Schußregen auszulösen.

Wenn es Space Pilot während des Gefechts gelingt, einen Dreierverband feindlicher Raumschiffe abzuschießen, bekommt er Sonderpunkte. Bei 10000 Punkten hat er sich ein Freileben – besser, Freiraumschiff – erkämpft, bei 20000 ein weiteres, und so fort.

Gespielt werden kann alleine oder zu zweit; ein Joystick ist nicht unbedingt erforderlich.

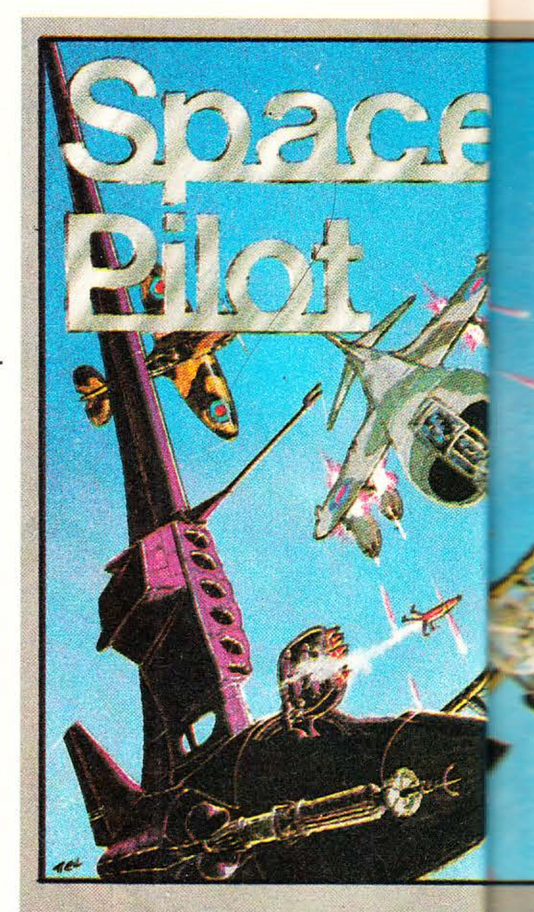
PILOT X

Oranges Raumschiff auf dem Weg zwischen lila Felsen: Pilot X soll ein außerirdisches Höhlensystem erforschen. Eine nicht ganz ungefährliche Aufgabe: Die Höhlenwände treten immer enger zusammen, so eng sogar, daß Pilot X manchmal die beiden Außenkapseln ausfahren und sie durch Nebengänge steuern muß, um sein Schiff vor dem Zerschellen zu bewahren. Er braucht hier zwar nicht unbedingt einen Joystick, dafür aber umso mehr Geschicklichkeit, gute Reaktionsfähigkeit und Übung.

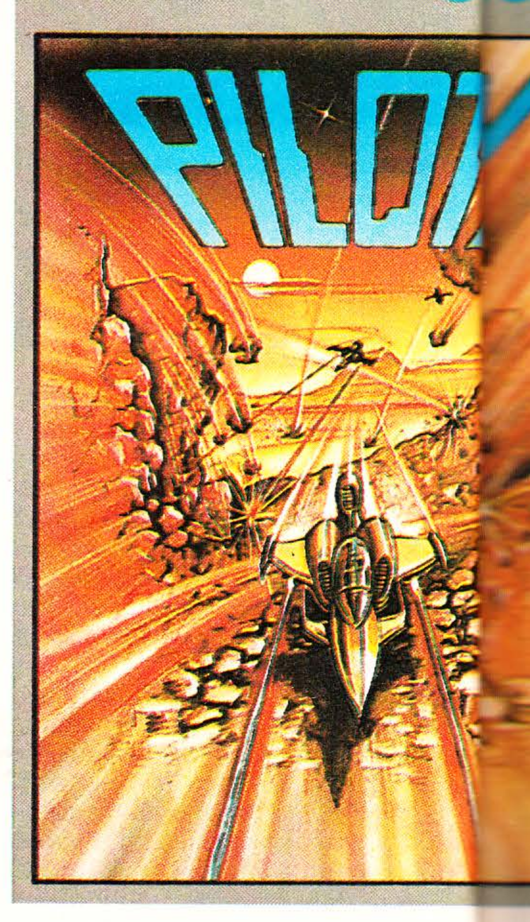
Um sich besser konzentrieren zu können, kann Pilot X (mit f1/f2) die Musikberieselung abschalten. Und manchmal legt er auch mit 'help' ein Päuschen ein: Immer wieder Felswände, fliederfarben, dieses immergleiche Tempo – das kann schon ziemlich ermüdend sein.

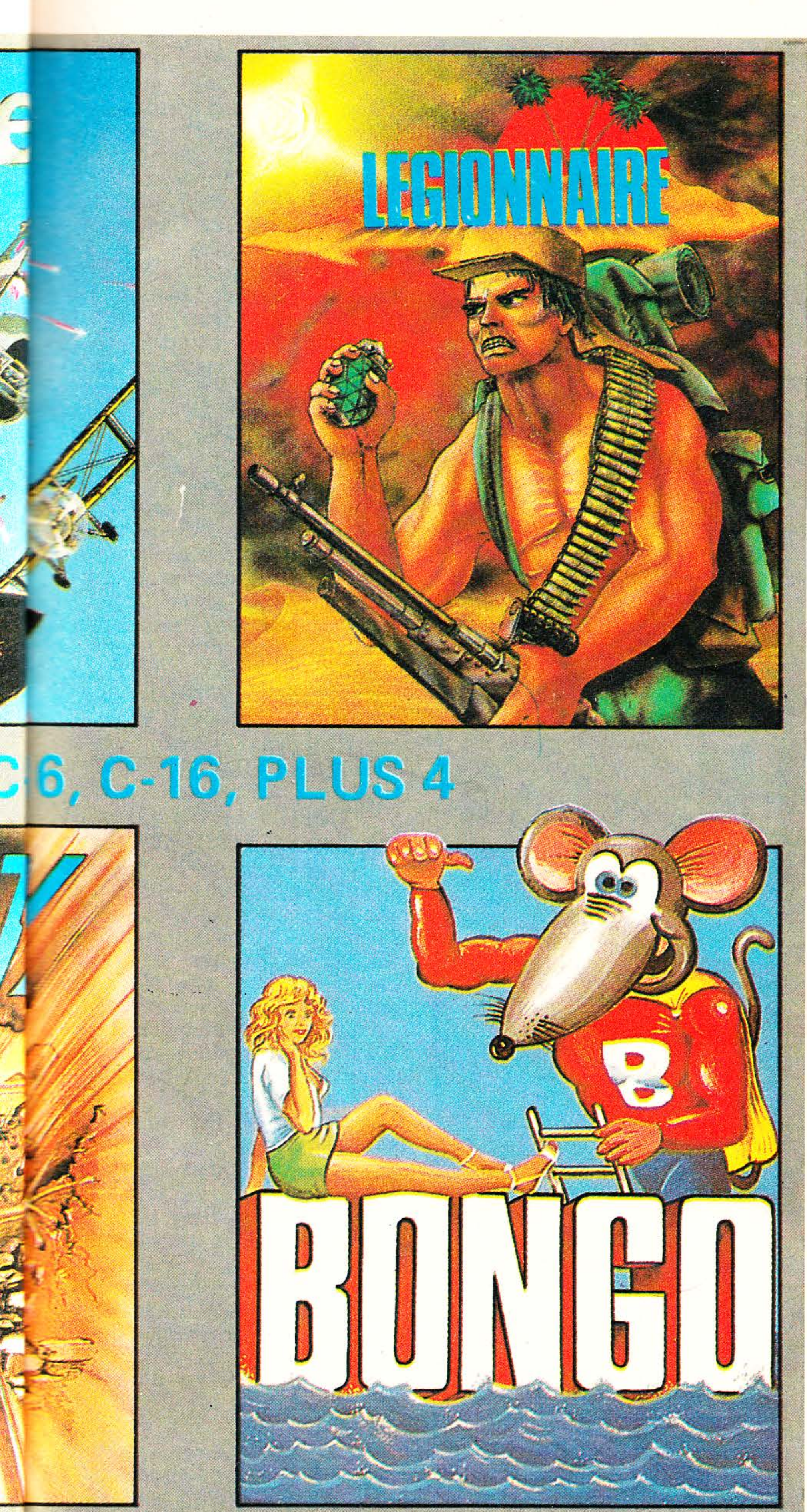
LEGIONNAIRE

Sand, Palmen und hie und da ein paar Büschel Steppengras – nein, Du bist nicht im Urlaub am Strand von Mallorca. Sondern mitten in irgendeiner Wüste, die sich durch einfachen Joystick-Knopfdruck ins Schußfeld wieselflinker blauer Männchen verwandelt. Du fängst an zu rennen, läufst und läufst, um den Kugeln der Gegner auszuweichen. Palmen und Sandhügel stehen Dir teils im Wege herum, teils suchst Du Deckung hinter ihnen und siehst dich plötzlich in die Enge getrieben. Aber Du mußt weiter: du hast eine wichtige Nachricht für das Hauptquartier. Und immerhin hast Du sechs Leben. Wenn Du geschickt genug bist, erreichst Du innerhalb dieser Galgen-



C 6





frist das offene Feld. Da kannst Du Deinen Gegnern endlich Kontra bieten: Du wirfst den Joystick (ohne den geht's nicht) an, drehst Dich frontal zum Feind und drückst auf den Joystickknopf, knallst sie ab, einen nach dem anderen. Dann kommst Du in eine neue Landschaft, der ersten reichlich ähnlich: Wieder tigerst Du durch ein scrollendes Wüstenbild. Zwischendurch stößt Du auf Flüsse, da mußt Du drüber. Und ab und zu schießt Du immer mal so ein aggressives blaues Männchen ab. So kämpfst Du dich durch sieben Landschaften bis zum Hauptquartier, wo Du dich etwas ausruhst – um Dich neuen Aufgaben – z.B. den anderen Spielen dieser Kingsoftdiskette oder -kassette – zuzuwenden.

BONGO

Prinzessin Legia ist blond, schön und – ganz wie es sich für so ein Königskind gehört – unglücklich: Der böse Zauberer Batrus hat fieserweise ihre Diamanten gemopst. Jetzt sind sie irgendwo in dem winkeligen Häuschen des Zauberers verstreut. Das ist natürlich ein unhaltbarer Zustand: Bongo, das Supermäuschen, schwingt sich heldenhaft durch sechs Bilder, um die Klunker wieder einzusammeln. Dabei muß er aufpassen, daß ihn der Schatzhüter des Zauberers nicht mit brutalen Kinnhaken ins Aus befördert. Aber Bongo hat einen Vorteil: Während der Bösewicht mit dem fürchterlichen Panzerknacker-Face bei der Verfolgungsjagd nur die Leitern benützen kann, hopst Bongo über Trampolins, benützt Rutschbahnen, Luftturbinen, Fließbänder, Transporter genauso wie Leitern.

Außerdem hat jeder Bongo seinen eigenen Herrn. Und der kann selbst entscheiden, wie hart sich sein Bongo bei seiner Aufgabe tut (wahlweise Level 1 - 4), wieoft er innerhalb eines Bildes von des Zauberers Handlanger gekillt wird (4 - 9 Leben) und in welchem Hexenhäuschen er seine Sammelei beginnt (Screen 0 - 5).

Nun aber das Besondere an diesem Spiel 'Bongo'.

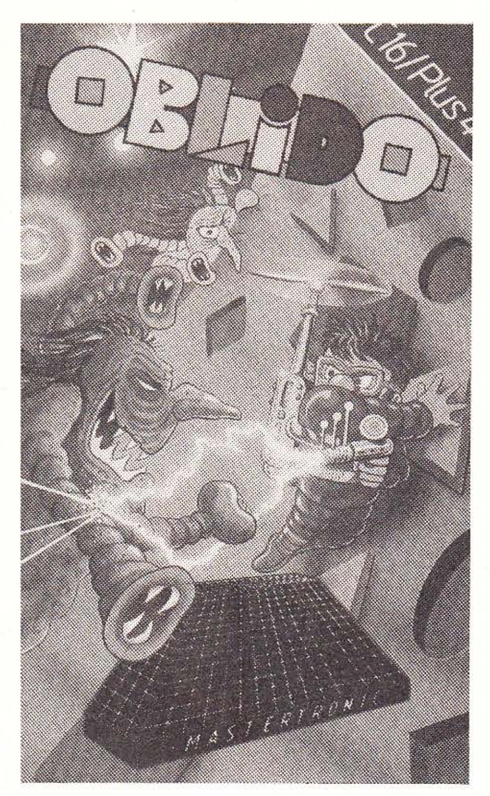
Programmacher Udo Gertz hat sich dazu ein nicht ganz gewöhnliches Special einfallen lassen: Den 'Construction Set'. Da kann man die Häuschen nach den eigenen

Vorstellungen zusammenbasteln und – auf einer Extradiskette oder -kassette – abspeichern: Mit dem Programm 'put objects' lassen sich Leitern, Rutschen und andere Fortbewegungsmittel verschieben, hinter anderen verstecken oder ganz aus dem Bild räumen. Aber Vorsicht: Schnell sind Situationen konstruiert, in denen Bongo keine Chance hat, den Schläger des Zauberers auszutricksen und die Diamanten zu ergattern.

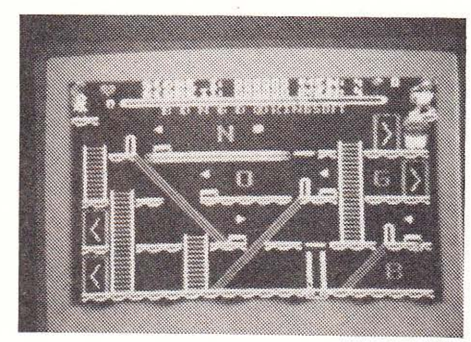
Insgesamt können bis zu 60 selbstentworfene Bilder gespeichert werden. Das geht so: Mit 'next picture' (0 - 5 Möglichkeiten) screen wählen und Bild entwerfen. Dann 'load save' anwählen, 'name picture' (0 - 9 Möglichkeiten) bestimmen und abspeichern. Allerdings: Für den noch wenig versierten Spieler mangelt's der Beschreibung an Präzision: Der Weg zum Vollgenuß des 'Bongo Construction Set' ist ihm mit geduldigen Versuchen gepflastert.

OBLIDO

Probleme mit Ordnung? Schlechte Nerven? Endlose Aufräumaktionen ohne sichtbaren Erfolg? Mastertronic gibt Gelegenheit zum Üben: Da muß man je neun rote, grüne, gelbe und blaue Kartons mittels Joystick oder Tastatur in das farblich dazupassende Zimmer schieben. Wer schon mal umgezogen ist, weiß, was dann zwingenderweise passiert: Die Schachteln verkeilen sich ineinander, bleiben in winkeligen Gängen stecken, so daß man selbst nicht mehr vorbei kommt, und zuguterletzt drängt die Zeit, weil, sagen wir mal, der Nachmieter schon vor der Türe steht. Und kaum hat man, wie man glaubt, die ersten Erfolge errungen, da kommen plötzlich so kleine Biesterchen namens 'sinners' – vielleicht die Nachbarskinder? – und halten einen von der Arbeit ab. Da muß man schnell sein und ihnen durch Druck auf den Joystickknopf eins drübergeben – aber rechtzeitig, sonst haben sie sich schon an einen rangehängt und klauen einem wertvolle Zeit, hier gemessen in Bonuspunkten. Und die sind Nullkommanichts aufgebracht: Die gerade mühsam in die richtige Richtung beförderten Schachteln fliegen durcheinander und die ganze Sysiphusarbeit beginnt von vorne. Es ist beileibe keine Übertreibung, wenn die Beschreibung verspricht: 'The eternally frustrating strategy game'.







Wer behauptet denn, daß die "Kleinen" nicht auch gute Grafiken erzeugen können, wenn der Programmierer was kann? Viele Spiele beweisen, daß sich die Serie der mit dem Basic 3.5 ausgestatteten Commodores nicht hinter grösseren verstecken muß.

DEMOLITION

Weg mit den grauen Mauern! Kingsoft macht's möglich: Während Stadtväter Abrißbirnen ordern, setzen wir uns an unsern C 16, C 116 oder C 4 und spielen Demolition. Der Joystick genügt uns, um wahlweise alleine oder zu zweit mittels Ball Stein für Stein abzuschiessen. Rote Steine sind ballfest – geschickte Spieler nutzten sie wie beim Billard als 'Bande'. In den höheren Levels – insgesamt immerhin 36 – kommen noch andere Hindernisse – nennen wir sie Spezialsteine – hinzu, die nur mit ganz bestimmten Tricks kleinzukriegen sind. Z.B. sind dann drei Treffer nötig, um die Mauer zu zerstören.

Und wenn den städtischen Abrißliebhabern widerspenstige Stadtratsgrüne die Pläne platzen lassen, so bedroht uns hier jeweils nach vier Bildern ein Gesicht, das seinerseits auf uns schießt. Bei Abschuß heißt's leider zurück zu Level 1. Wer jetzt müde wird, kann sich vom Ballgeräusch (Noise=on) etwas animieren lassen. Und Ungeduldige können die Ballgeschwindigkeit erhöhen – erstaunlicherweise räumen manche so trotz des erforderlichen höheren Reaktionsvermögens schneller ab. Zaudern ist dem Programm sowieso zuwider. Wenn man, bereits spielbereit, ausharrt und auf bessere Zeiten wartet, zeigt einem das Programm von sich aus, wie meisterhaft man sich in den verschiedenen Bildern schlagen kann: Auf diese Weise bekommt wenigstens auch der weniger Geübte mal ein paar von den höheren Levels zu Gesicht. Aber Vorsicht Glotzensüchtige: Das Zuschauen macht Heidenspaß. Da könnte man glatt das Spielen vergessen. Taktisch klug, daß die Vision verschwindet, bevor der letzte Stein exekutiert ist: Das frustet – und man spielt doch lieber selbst. Für 19.95 (Diskette oder Kassette) bekommt man ein wirklich hervorragend gemachtes Spiel, das den Vergleich mit wesentlich teureren Amigaspielen nicht zu scheuen braucht und eine echte Bereicherung der eigenen Spielesammlung darstellt. *sad.*

ZAUDERN IST DIESEM PROGRAMM EIN GREUEL

DREI AUF EINEN STREICH

Eine Sammlung von drei Spielen bietet CSJ Computersoft Jonigk unter dem Namen „CSJ-Games 1" an, für die Computer C16/116/Plus 4. Eines davon, nämlich „Sun Street" ist uns hierbei bereits als „Paper Boy" aus dem C16 Sonderheft 2/87 bekannt. Wir waren gespannt auf die zwei übrigen.

Was bei den Gebrüdern Grimm der Griesbreiberg, ist bei Programmautor Tom Fresen das Erdbeerfeld: Da muß Honny sich durchfressen. „Wie schön", möcht' man da begeistert rufen – aber leider sind die Erdbeeren ohne Sahne, und außerdem werden sie von allerhand grünen Ungeheuern oder Riesenblattläusen bewacht, die – kaum hat er sich an so eine Beere rangepirscht – ihrerseits versuchen, den Räuber zu vernaschen. Dabei hat Honny keine Zeit zu verschenken: Wenn er nicht innerhalb von 15 Sekunden etwas zu knabbern hat, ist seine Energie – links im Bildschirm als Zickzacklinie abzulesen – dahingeschmolzen; er stirbt jämmerlich zitternd vor Angst und Schwäche. Aber macht nichts, er hat ja nicht nur ein, sondern vier Leben. Und mir nichts dir nichts, ohne noch vom erschöpften Spieler durch „fire" gerufen worden zu sein, steht er schon wieder auf der Matte, sprich: im Erdbeerfeld.

THE JOLLY GOOD FELLOW

Will er nicht wieder an Kraftlosigkeit zugrundegehen, hat er nur eine Chance: Er muß warten, bis die grünen Monsterchen weit genug weg sind, nämlich im nächsten Feld. Von dort aus bekommen sie nämlich nicht so leicht mit, wenn Honny sich an den Früchten vergreift. Aber je weiter sich Honny durch die Plantage mampft, desto schwieriger wird das: die Felder werden nämlich immer größer, so daß schließlich ein Monster nur noch ein Feld bewacht: Da muß Honny wirklich geschickt vorgehen. Der leckere Lohn: Hat er die erste Plantage abgeerntet, ohne selbst gefressen zu werden,darf er im nächsten Spiel in die nächst größere Plantage mit noch mehr Erdbeeren, aber auch mit noch mehr gefährlichen Wächtern. So weit dürfen Honny freilich nur ganz wenige führen. Das Problem: Unser Held reagiert gerade auf ungeübte Hände am Joystick etwas, sagen wir, trotzig. Da setzt er sich dann einfach in seine Ecke links über der ersten Erdbeere und wartet, bis seine Kraft verbraucht ist. Wenn da dann zwei Trotzköpfe – einer vor und einer auf dem Bildschirm – aufeinandertreffen, ist dem Game wahrscheinlich eher staubiges Siechtum in der Diskettenkiste beschert. Aber halt, ich vergaß ja: Auf der Diskette haben wir ja noch zwei andere Spiele: „Danger Diamonds" (schafft garantiert schnelle Erfolgserlebnisse!) und „Sun Street".

DANGER DIAMONDS

Drei Leben a 45 Sekunden – wenn Jim in seinem kurzen Erdendasein zu etwas kommen will, hat er keine Zeit zu verlieren: Er ist losgezogen, um Diamanten zu suchen – die Diamantmine hat er schon gefunden. Jetzt gilt es nur noch, die Diamanten schnell einzusammeln und dabei den von hinterhältigen Ganoven gelegten Tretminen oder Steinschlag auszuweichen. Wenn Jim während eines Lebens zehn Diamanten ergattern kann, rückt er in den nächsthöheren Level auf. Da gibt's zwar noch mehr Diamanten, aber auch noch mehr Tretminen und Steine, die Jim bedrohen. Sobald unser lonely Cowboy auf Knopfdruck (fire) gestartet ist, muß man ihn mittels Joystick schnell durch das gefährliche Gebiet dirigieren: Tückischerweise kann es durchaus passieren, daß sich unvermutet ein Stein löst und Jim auf den Kopf fällt.

Trotz dieser kleinen Zwischenfälle ist das Game auch für den blutigen Anfänger leicht zu bewältigen.

Und wer „Danger Diamonds" satt hat, kann ja aus dem Menü der Diskette noch „Sun Street (The Newsboy)" oder „The Jolly Good Fellow" wählen.

Alle drei Solospiele kosten 32,– DM auf Kassette bzw. 49,– DM auf Diskette.

sad

BUMM...

Wir kennen's aus langweiligen Schulstunden: „Schiffchen versenken" gibt es jetzt von „Mastertronic" C-16- und P-4-gerecht auf Kassette. Irgendwo in der Nordsee, zwischen Jütland und den Shetlandinseln, machen sich zwei Ölgesellschaften die letzten Ölvorkommen streitig. Und schon haben wir den Salat, sprich, Seekrieg: Die diplomatische Verständigung hat offensichtlich mal wieder nicht geklappt. Beide Parteien stehen sich bis an die Zähne hochgerüstet, kontrastreich rot und blau, gegenüber, um den andersfarbigen Bohrtürmen, Schiffen, U-Booten, dem Flugzeug und dem Hubschrauber den Garaus zu machen. Einzelkämpfer antworten an dieser Stelle auf "play me" mit "Y",

BATTLE

um gegen den Computer zu fighten; zwei Spieler wählen "N" und hetzen gegeneinander. Von einem Blitzkrieg kann dabei freilich keine Rede sein: Den meisten Teil der Zeit beansprucht die „Input-Phase", während derer erst die Fahrtrichtung der einzelnen Schiffe bestimmt und dann der Befehl zum Schießen eingegeben wird. Das erfordert – bei all den fitzelkleinen Schiffchen – ein gerüttelt Maß an Aufmerksamkeit und gute, computerfeste Augen. Und es dauert geraume „Input"-Runden, bis die Gegner trotz massivem Kampfeswillen überhaupt erst auf Schußentfernung aneinander herangekommen sind. Aber eingefleischte Tüftler haben vielleicht

...BUMM

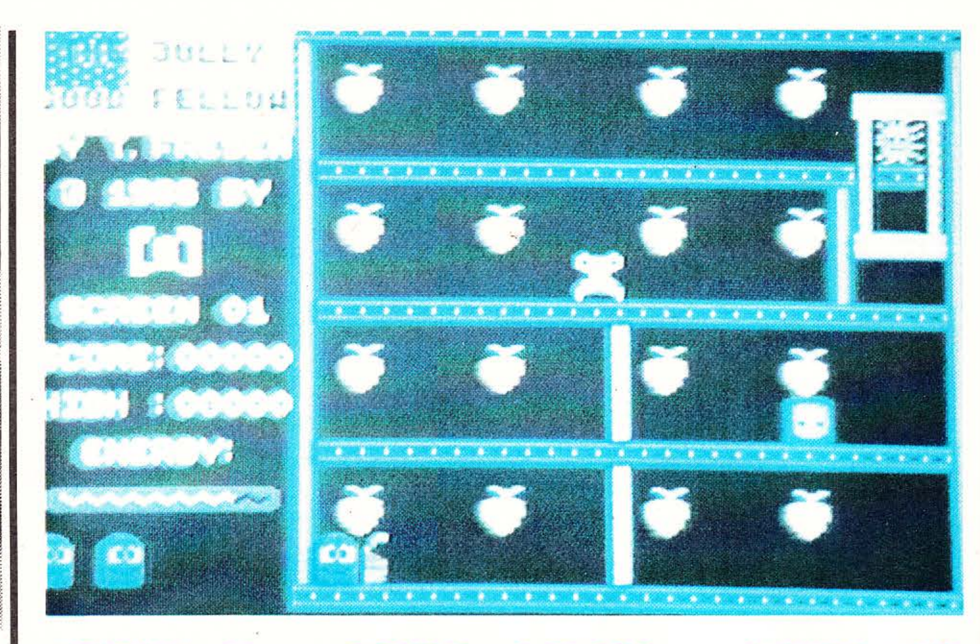
ihre Freude daran, sich durch die Input-Runden zu rackern und sich etwa mit ihrem U-Boot mittels aufwendiger Manöver an den Feind heranzupirschen. Nur Vorsicht beim Tauchen: Da rempelt man schon aus Versehen an einen Bohrturm oder eine Insel und geht dabei unter. Auch wer ungeduldig wird und zu früh zu feuern beginnt, zieht den Kürzeren: Jede Partei verfügt nur über begrenzte Vorräte an

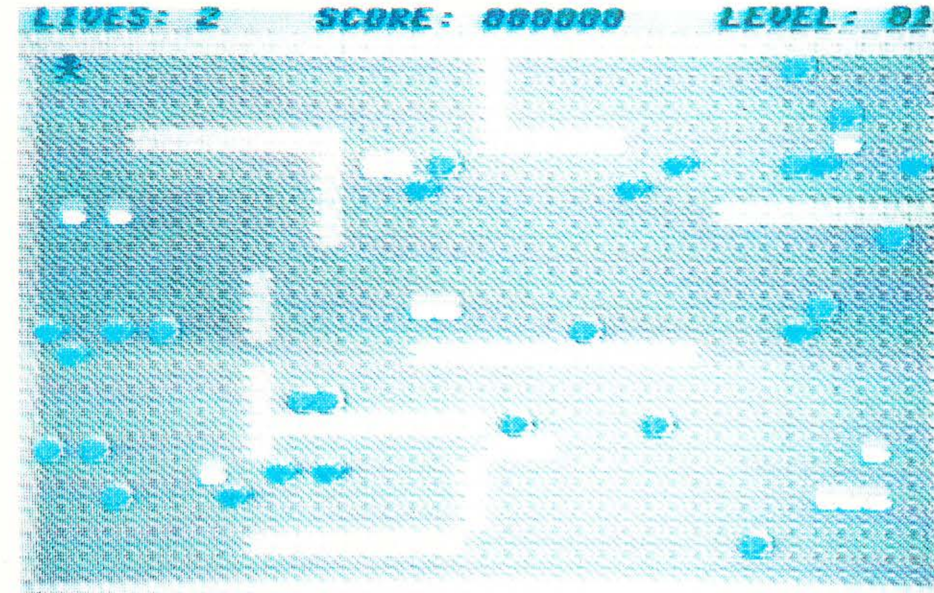










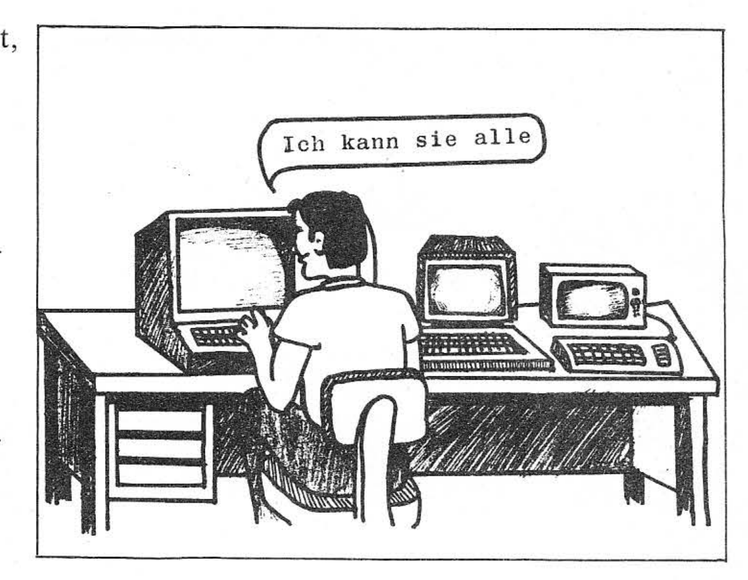


Raketen, Kanonen, Torpedos und Bomben. Was soll's – ungezieltes Losballern, so lehrt schon die Geschichte, hat sich noch nie so recht bewährt. Da trifft man im Kampfeseifer schon mal ein eigenes Schiff. Wenn's auch kein ekelhaftes Blutvergießen gibt – schmerzlich ist es trotzdem, wenn man die eigenen Mannschaften unter feuerrot angestrahlten Wölkchen am unteren Bildschirmrand versinken sieht, und der Feind am Horizont lacht sich von weitem ins Fäustchen.

Und wenn man dann endlich in Vernichtungsnähe am Feind dran ist? – Es mag trotzdem keine rechte Kampfesfreude aufkommen: Zu oft muß man hier die Spielanleitung wälzen, um sich zu vergewissern, welches Schiff mit welcher Waffe auf welche Entfernung welches Objekt vernichten kann. Einziger Ausweg: Eingabemöglichkeiten auswendiglernen. Wen das abschreckt, oder wem die 5,90 DM für die von Rushware erhältliche Kassette zuviel sind, mit Papier und Bleistift macht die Versenkerei in diesem Fall genausoviel Spaß.

EIN COMPUTER WAS IST DAS?

Wer alle Möglichkeiten seines Computers voll ausschöpfen möchte, bemerkt sehr schnell, daß er mit Basic nicht sehr weit kommt. Weiterführende Aufsätze in Computerzeitschriften setzen bereits soviel an Grundlagenwissen voraus, daß man nur frustriert aufgeben kann. In diesem Bericht schließen wir diese Lücke und zeigen, wie ein Computer funktioniert. Sie erfahren Wissenswertes über seine Bausteine und lernen, ihn mittels Assembler, Compiler und Interpreter zu programmieren. Damit steht Ihnen ein Programmiersystem zur Verfügung, mit welchem Sie jeden Computer mit Komfort programmieren können.



Wie ein Computer aussieht, dürfte fast jedem, jedenfalls aber unseren Lesern, bekannt sein. Darüber, was ein Computer eigentlich seinem Wesen nach ist, herrschen viele, oft falsche, Vorstellungen. Manche meinen, ein Computer sei gescheit oder klug, weil er so vieles kann, manche betrachten ihn auch fast wie ein intelligentes Wesen.
Wenn ein Computer aber intelligent ist, dann jedenfalls nur in dem Sinne, wie man manche Bücher auch als schlau bezeichnet. Nicht das Papier des Buches ist schlau, oder die Druckerschwärze, sondern höchstens der Autor, der mit Einfallsreichtum und Ausdauer seine Ideen in lesbare Form gebracht hat. Die Information in einem Buch ist dafür gedacht, vom Auge eines Menschen wahrgenommen zu werden. Die Datentechnik tut sich dagegen mit Daten, die auf Magnetträger gespeichert sind, wesentlich leichter.

DER DATENRECORDER ALS STEUERGERÄT

Normalerweise werden die Signale eines Kassettenrecorders an einen Lautsprecher gesandt, der sie dann als Sprache oder auch als Musik für das menschliche Ohr hörbar macht. Jedoch sind auch andere Verwendungsarten denkbar. Als Datasette für einen Computer ist der Recorder vielen bekannt. Wir könnten ihn auch als Massenspeicher für eine elektronische Schreibmaschine benützen, oder Text auf einen Bildschirm ausgeben. Wir könnten ein sogenanntes automatisches Klavier mit ihm bedienen, oder die Bewegungen eines Greifarmes mit ihm steuern.

WAS MACHT DEN COMPUTER ZUM COMPUTER?

Ein Kassettenrecorder oder auch ein Filmprojektor haben etwas gemeinsam. Sie spulen ihre Aufzeichnungen ab, und zwar ganz genauso, wie diese aufgenommen wurden. Wenn wir uns nun vorstellen, daß ein Greifarm vielleicht nur ein paar Bewegungsmuster benötigt, um verschiedene Aufgaben zu erledigen, daß aber so ein Bewegungsmuster oft aus einer Unzahl kleinster Einzelbewegungen zusammengesetzt ist, die alle gesteuert werden müssen, so wird uns vielleicht bewußt, was ein Computer einem Datenabspielgerät voraus hat, nämlich die Selbststeuerung.

DIE ERFORDERLICHE ERGÄNZUNG

Auf einem Recorder, der sich selbst steuern kann, braucht nun nicht jedesmal eine ganze Steuersequenz für ein bestimmtes Bewegungsmuster abgespeichert sein, sondern lediglich die Befehlsfolge, die notwendig ist, um das Band auf die Stelle zu spulen, wo sich die entsprechende Sequenz befindet. Allerdings gibt es eine Schwierigkeit. In dem Augenblick nämlich, wenn das Band beginnt, sich zurückzuspulen, wird das Steuerprogramm für den Rückspulvorgang unterbrochen und das Unternehmen ist mißlungen. Abhilfe schüfe ein zweiter Recorder, der dem ersten diese Aufgabe abnähme.

MACRO- UND MICROPROGRAMM

Natürlich ist es etwas aufwendig, einen Recorder zu steuern. Mit einer Musikbox braucht man nur zwei Tasten zu drükken und schon ist die richtige Platte angewählt. Anstelle eines zweiten Recorders stellen wir uns daher besser eine Musikbox mit vielen Platten vor. Ein Code auf dem Band reicht nun aus, eine ganze Codesequenz anzusteuern. Ein Macrocode, sozusagen, generiert eine Folge von Microcodes. Die Programme auf den Platten nennen wir daher „Microprogramme" und das Programm auf dem Band „Macroprogramm".

DER BEFEHLSZYKLUS

Haben wir bei einem Datenabspielgerät einfach das Band abspielen lassen können, so hat nun eine gesonderte Regelung zu erfolgen. Macroprogramm und Microprogramme müssen gegenseitig Rücksicht nehmen. Erst wenn das gewählte Microprogramm abgearbeitet ist, darf mit dem Macroprogramm weitergefahren werden. Die Steuerung dieses Vorganges übernimmt nun ein Microprogramm, das folgendes tut:

1 Code lesen
2 Code in Befehlsregister ablegen
3 Microroutinenaufruf
4 Weiter mit 1



Drei Microprogramme erlauben die Bandsteuerung. Der Befehl "JMP" (Jump), gefolgt vom gewünschten Zählerstand, verstellt das Band. Vorgänge, die stets gleich bleiben, sich also ständig in derselben Weise wiederholen, lassen sich automatisieren. Mit dem JMP-Befehl lassen sich prima Wiederholungen realisieren:

```
:BEGIN  .........
        .........
        .........
        .........
        .........
        JMP BEGIN
```

Der Befehl "CALL" verstellt ebenfalls das Band. Zusätzlich merkt er sich aber vorher den alten Zählerstand. Mit einem weiteren Befehl "RET" kann nach Ausführung einer Routine an anderer Stelle wieder die ursprüngliche Bandposition hergestellt werden:

```
:BEGIN  CALL GETIN
        CALL BSOUT
        JMP BEGIN

:GETIN  ............
        ............
        ............
        RET

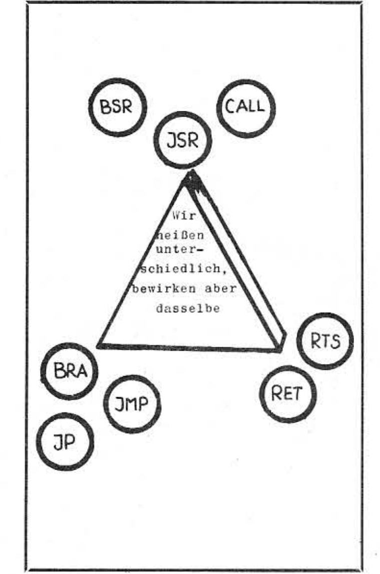
:BSOUT  ............
        ............
        ............
        RET
```

Wenn die hier angedeutete Routine GETIN dafür sorgt, daß ein Zeichen von der Tastatur eingelesen wird, und wenn wir weiterhin annehmen, daß BSOUT die Bildschirmausgabe bewirkt, so sorgt unser Programm nun dafür, daß jeder Tastendruck auf den Bildschirm geschrieben wird.

ASSEMBLER-MNEMONICS UND LABELS

In unseren Programmbeispielen verwendeten wir keine Computerdaten, sondern Namen, denn Zahlen sind nichtssagend, besonders, wenn unterschiedliche Computer unterschiedliche Zahlen für dieselbe Routine benützen. Ein Begriff dagegen ist weit griffiger. Assembler-Mnemonics sind abgekürzte Begriffe, mit denen üblicherweise Maschinenbefehle bezeichnet werden. Leider gibt es kleine Abweichungen in der Begriffsgebung bei unterschiedlicher Assembler- und Bausteinentwicklern, wie Sie es folgender Gegenüberstellung entnehmen können:

CPUs	68000	8086	Z80	6502 /	BASIC
	BRA	JMP	JP	JMP	GOTO
	BSR	CALL	CALL	JSR	GOSUB
	RTS	RET	RET	RTS	RETURN



Basic ist kein Microprozessor, sondern eine Programmiersprache, jedoch auch hier sind die besprochenen Routinen vertreten.
Ein Assembler ist ein Programm, das ein als Text in Assembler-Mnemonics geschriebenes Programm in die Daten übersetzt, die der Computer versteht. Im Assembler Programm brauchen nicht die Adressen (Zählerstände) angegeben werden, auf welche nun der CALL oder JMP springen soll, das wäre manchmal eine mühsame Rechnerei. Wenn man nachträglich ein paar Zeilen einfügen möchte, verschöben sich alle nachfolgenden Adressen und man müßte diese alle neu berechnen.
Beim Assembler braucht man nur Marken (Labels), zumeist Namen am linken Rand und mit Doppelpunkt zusätzlich besonders kenntlich gemacht. Um den Rest brauchen wir uns dann nicht mehr zu kümmern, denn das Berechnen der Adressen besorgt der Assembler.
Mit den besprochenen Routinen bringt ein Computer erstaunliche Kunststücke fertig. Anstatt das Band zu verstellen, könnten wir uns unsere Routine auch anders definieren, z.B. es sollen die Bänder gewechselt werden.

WO BLEIBT NUN DIE SOGENANNTE INTELLIGENZ?

Von einer Intelligenz des Computers war bislang noch nichts bemerkbar. Worin äußert sich Intelligenz? Doch darin, daß man sich richtig in verschiedenen Situationen verhalten kann, wenn man z.B. die richtigen Antworten weiß, so mögen manche meinen.
Ein Computer nun hat die Möglichkeit, auf unterschiedliche Situationen unterschiedlich zu reagieren. Ob allerdings richtig, das hängt nicht von einer Intelligenz des Computers, sondern von der des Programmierers ab.

BEDINGTE SPRÜNGE

Mehrere Microroutinen, die "BRANCH"-Routinen, ermöglichen den bedingten Sprung. Ob ein Sprung stattfindet, hängt nun vom Zustand eines gewissen Registers, dem sogenannten Statusregister, ab. Solch ein Statusregister beinhaltet verschiedene "Flags", die sogenannten Statusflags. Wir stellen uns am besten das Statusregister als Schalterleiste vor, wobei die Flags die Schalter darstellen. Die Schalter können nun auf "Ein" oder auf "Aus", bzw. auf 1 oder 0 stehen. Angenommen, es soll ein Sprung erfolgen, wenn Flag0 den Wert 1 hat, so erreichen wir es mit:
BF0 : Branch if Flag0=1
Sollte Flag0 den Wert 0 haben, so wird der Sprung nicht ausgeführt, sondern einfach mit dem darauffolgenden Befehl fortgefahren.
Folgender Befehl würde das Gegenteil bewirken, nämlich einen Sprung, wenn Flag0=0 und Unterlassung, wenn Flag0=1:
BNF0: Branch if not Flag0=1

Hierzu ein Beispiel:
Die Macroroutine "MESSEN" mißt die Raumtemperatur. Als Ergebnis dieser Messung werden Flag0 auf 1 gesetzt, wenn es zu kühl ist. Wenn nicht, so habe Flag0 den Wert 0. Wenn es zu kühl ist, soll geheizt werden, wenn nicht, dann ist die Heizung auf "Aus" zu setzen. Die Routine "HEIZEN" besorgt das Heizen, die Routine "KUEHLEN" das Gegenteil. Unser Programm, das nun heizen oder kühlen soll, sieht nun folgendermaßen aus:

```
:REGEL0 CALL MESSEN
        BF0  REGEL1
        CALL KUEHLEN
        JMP  REGEL0
:REGEL1 CALL HEIZEN
        JMP  REGEL0
```

Genauso richtig könnte man auch schreiben:

```
:REGEL0 CALL MESSEN
        BNF0 REGEL1
        CALL HEIZEN
        JMP  REGEL0
:REGEL1 CALL KUEHLEN
        JMP  REGEL0
```

CHIPS STATT GRIPS

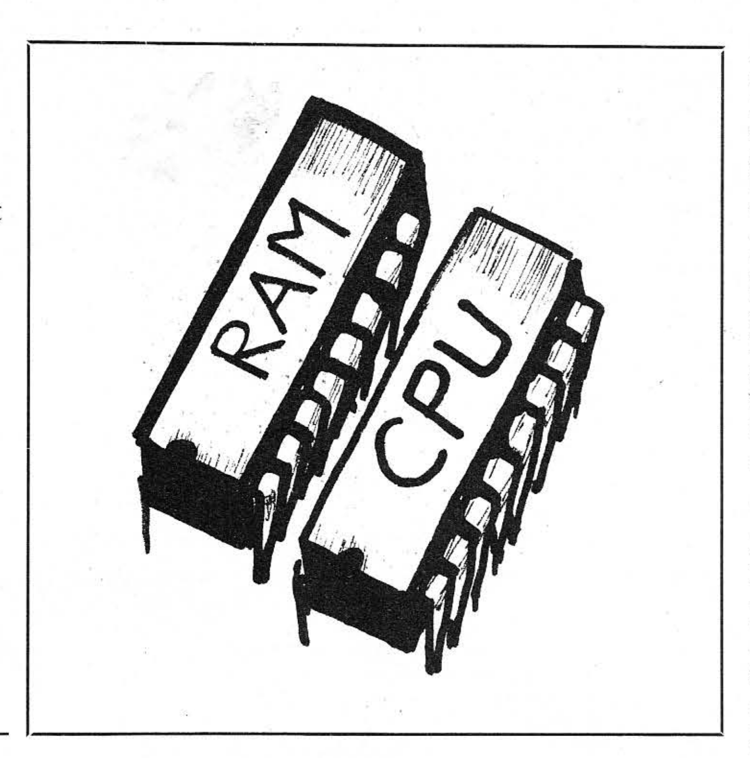
Das, was einen Computer ausmacht, hat nicht mit Grips zu tun, mit einfachen Geräten, wie z.B.

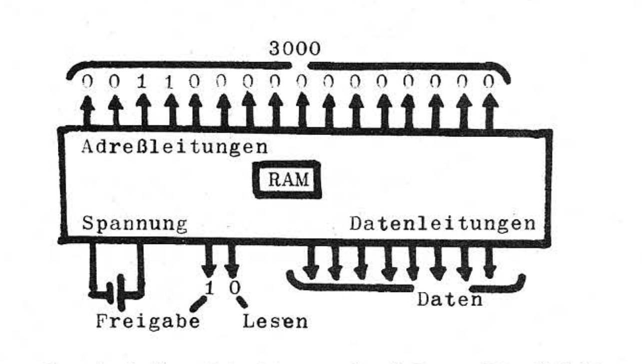
einem Kassettenrecorder und einem musikboxähnlichem Apparat wäre er bereits zu verwirklichen. Jedoch was den Computer besonders auszeichnet ist seine Schnelligkeit, mit der er Programme ausführen und Datenzugriffe vornehmen kann. Daher wollen wir unsere etwas antiquiert anmutenden Vorstellungen beiseite legen und uns mit den Bausteinen auseinandersetzen, aus denen ein Computer tatsächlich besteht, den sogenannten Chips. Uns interessiert nun nicht, wie diese im Detail hergestellt sind, sondern das Funktionsprinzip. Wir lassen nun RAM und CPU die von Kassette und Platten einnehmen.



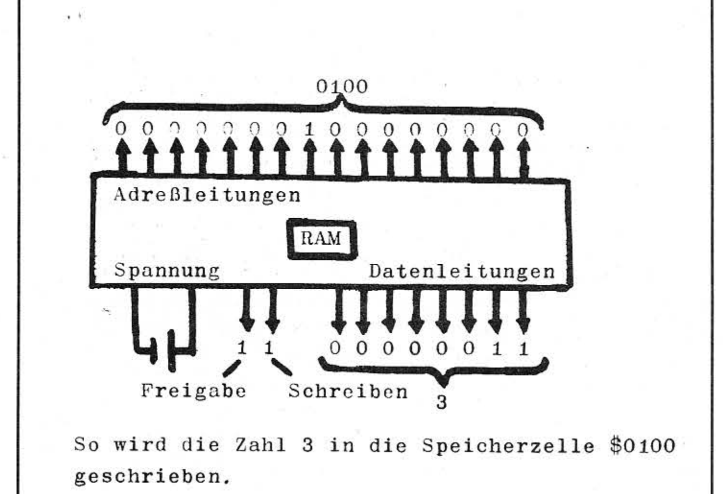
RANDOM ACCESS MEMORY

Ein Kassettenrecorder hat einige Nachteile. Er ist relativ langsam. Technisch aufwendig wäre wohl eine exakte Bandansteuerung zu realisieren, mit welcher jede beliebige Stelle exakt angesteuert werden könnte. Das Vor- und Zurückspulen nimmt erhebliche Zeit in Anspruch. Ein Band ist eben mehr dafür gedacht, der Reihe nach gelesen zu werden. Ein Speicherbaustein, der einen wahlfreien Zugriff erlaubt, ist das Optimale. Speicher mit wahlfreiem Zugriff heißt in englisch "RANDOM ACCESS MEMORY", abgekürzt auch RAM genannt. An so einem Baustein sitzen nun verschiedene Leitungen, nämlich Stromversorgung, Adreßleitungen, eine oder mehrere Datenleitungen, und wenigstens zwei Steuerleitungen, nämlich eine für die Wahl von Lesen oder Schreiben und eine für das Freigabesignal. Um nun Daten aus dem RAM zu lesen oder auch hineinzuschreiben, brauchen nur entsprechende Spannungen an den Leitungen anzuliegen und schon ist der Vorgang möglich.





So wird der Inhalt von Speicherzelle $3000 gelesen.



So wird die Zahl 3 in die Speicherzelle $0100 geschrieben.

An die Stelle unserer Musikbox tritt die CPU, die Zentraleinheit, um die sich alles dreht, auch oft als Herz des Computers bezeichnet. Da das RAM, wenn es nicht abgefragt wird, von selbst überhaupt nichts tut, die CPU aber ständig – wir erinnern uns an den ablaufenden Befehlszyklus – aktiv ist, sieht man sie als den wichtigsten Teil des Systems an. Genausogut könnte man aber auch argumentieren, daß die Aktionen der CPU ja durch die im RAM stehenden Codes gesteuert werden, daß deshalb das Programm als geistiges Element die Seele der ganzen Sache sei.
Die Anweisungen, die im RAM stehen, werden eben von der CPU ausgeführt. Da wir uns nicht in philosophische Spekulationen verlieren wollen, ob nun dem Geist oder der Materie die führende Rolle zukommt, oder wie es in der Computerterminologie heißt, der Software oder der Hardware, oder gar, ob nun das Huhn oder das Ei früher war, sehen wir uns die CPU etwas näher an. Rein äußerlich gleicht sie dem RAM. Auch die CPU hat Adreßleitungen, Datenleitungen, Steuerleitun-

BEFEHLSZÄHLER BEFEHLSREGISTER

gen und eine Stromversorgung. Der Unterschied zum RAM offenbart sich nicht in den sichtbaren als vielmehr in den unsichtbaren Dingen. Was die CPU haben muß, ergibt sich aus dem, welche Aufgaben sie zu erfüllen hat. Jedenfalls muß aber auch die CPU verschiedene Speicher, die in diesem Falle Register genannt werden, besitzen.
Diese Register dienen aber nicht dazu, einfach massenweise Daten irgendwo abzulegen, wie dies beim RAM der Fall war, sondern werden für spezielle Steuerungsaufgaben der CPU benötigt.



Da zum Lesen von Daten aus dem RAM deren Speicheradresse benötigt wird, muß die CPU sich ständig merken, wo sie den nächsten Befehl zu suchen hat.
Hierzu dient der Befehlszähler. Der Befehlscode, welcher nun gelesen werden kann, wandert in das Befehlsregister.
Der Befehlszyklus, den wir bereits besprochen haben, sieht nun so aus:

1 Befehlszählerinhalt auf Adreßbus
2 Lese/Schreibleitung auf Lesen
3 Freigabesignal senden
4 Datenbus an Befehlsregister
5 Befehlszähler erhöhen
6 Befehlsaufruf
7 weiter mit 1

Die Daten- und Adreßleitungen nennt man Datenbzw. Adreßbus. Der Befehlszähler wird um 1 erhöht, damit er nun auf den nächsten Befehl zeigt. Wird ein solcher Befehl abgearbeitet, so kann der Befehlszählerinhalt unter Umständen währenddessen auch weiter verändert werden, weil dem Befehlscode vielleicht weitere Parameter folgen.

SO WIRD GEJUMPT

1 Befehlszähler auf Adreßbus
2 L/S-Leitung auf Lesen
3 Freigabesignal
4 Datenbus an Daten register
5 Befehlszähler erhöhen
6 Befehlszähler auf Adreßbus
7 Datenregister in Befehlszähler H
8 L/S-Leitung auf Lesen
9 Freigabesignal
10 Datenbus auf Befehlszähler L
11 Rückkehr zum Befehlszyklus

Der Befehlszähler bekommt somit die dem Befehlscode folgende Sprungadresse übermittelt. Der anschließend folgende Befehlszyklus liest nun an der neuen im Befehlszähler vermerkten Adresse weiter.

Bei einem CALL ist die Rücksprungadresse zu vermerken. Mit einem normalen Register ist es hierbei nicht getan. Angenommen, es erfolgt der Aufruf einer Maschinenroutine durch CALL: Ein Rücksprung durch RET wäre dann ohne weiteres möglich. Was aber dann, wenn innerhalb der aufgerufenen Routine ein weiterer Routinenaufruf erfolgen würde? Die ursprüngliche Rücksprungadresse würde durch die neue überschrieben werden und ein Rücksprung in das anfänglich aufrufende Programm könnte nicht mehr erfolgen. Die Lösung bietet ein sogenannter Kellerspeicher, auch Stapel oder Stack genannt. Meistens wird hierzu ein RAM-Bereich verwendet. Ein Register, das Stapelregister, auch Stapelzeiger genannt, beinhaltet eine Adresse, in welche der nächste zu merkende Wert geschrieben werden soll. Nachdem ein solches geschehen ist, wird das Stapelregister entweder entsprechend erhöht oder erniedrigt, je nachdem, ob man sich zu einem Stapel- oder Kellerspeicher entschlossen hat. Dort kann dann ein erneuter Wert gespeichert werden und sukzessive weitere. Erfolgt ein RET (Return), so wird der Stapel einfach wieder entsprechend abgehoben und der Rücksprungwert wieder in den Befehlszähler gebracht.

ROUTINENAUFRUF DURCH CALL

1 Stapelregister erniedrigen
2 Stapelregister auf Adreßbus
3 Befehlszähler H auf Datenregister H
4 Befehlszähler L auf Datenregister L
5 Datenregister erhöhen
6 Datenregister erhöhen
7 Datenregister H auf Datenbus
8 L/S-Leitung auf Schreiben
9 Freigabesignal
10 Stapelregister erniedrigen
11 Stapelregister auf Adreßbus
12 Datenregister L auf Datenbus
13 L/S-Leitung auf Schreiben
14 Freigabesignal
15 weiter wie bei JMP

Bei den meisten CPUs liegt ein Kellerspeicher vor, weswegen wir in unserem Beispiel den Stapelzähler erniedrigt haben.

Datenregister
Adreßregister
Akkumulator
Statusregister
Indexregister
Stapelregister
Befehlsregister
Befehlszähler
CPU
Adreßleitungen
Datenleitungen
Steuerleitungen
Spannung

RÜCKSPRUNG DURCH RET

1 Stapelregister auf Adreßbus
2 Lesesignal setzen
3 Freigabesignal setzen
4 Datenbus an Befehlszähler L
5 Stapelregister erhöhen
6 Stapelregister auf Adreßbus
7 Lesesignal setzen
8 Freigabesignal setzen
9 Datenbus an Befehlszähler H
10 Stapelregister erhöhen
11 Rückkehr zu Befehlszyklus

DAS STATUSREGISTER

Das Statusregister enthält mehrere Statusflags. Jedes dieser Flags erfüllt eine spezielle Aufgabe. Wir haben sie einfach einmal durchnumeriert. In Wirklichkeit hat jedes einen besonderen Namen. Dieser kann aber erst in der nächsten Folge besprochen werden, weshalb wir vorerst bei unserer Terminologie bleiben. Zu den Branchbefehlen braucht vorerst einmal nichts mehr gesagt werden. Statt dessen machen wir mit den bisherigen Elementen unserer Sprache einige Übungen, welche wir auch an unserem Rechner ausprobieren können.

BEISPIEL: EINGABESCHLEIFE

Assembler

```
   ORG            $3000
   GETIN EQU      $FFE4
   BSOUTEQU       $FFD2

 :BEGIN CALL      GETIN
        CALL      BSOUT
        JMP       BEGIN
```

ORG $3000 sagt dem Assembler, daß er die Adressen so berechnen soll, daß das Programm an Adresse $3000 lauffähig ist. Diese Adresse wurde willkürlich gewählt, wir hätten auch $2000 nehmen können, oder eine krumme Zahl wie z.B. $24AF.
Die Routinen GETIN, welche eine Tastaturabfrage bewirkt, liegt bei unseren Commodore-Computern an Adresse $FFE4, die für die Zeichenausga-

die verantwortliche Routine an Adresse $FFD2. Mit der EQU-Anweisung wird dem Assembler gesagt, daß er, sofern er auf die solcherart definierte Konstante stößt, dafür den entsprechenden Wert einsetzen soll. Die Konstantendefinition durch EQU bietet den Vorteil, daß man nur die Definition am Programmanfang abzuändern braucht, wenn bei einem anderen Computersystem die bereits vorhandenen Maschinenroutinen anders im Adreßraum positioniert sind. BEGIN ist eine Labeldefinition. Da vorher noch kein einziger zu übersetzender Code stand, weist der Assembler dieser Marke den Wert $3000, welches unser Programm ist, zu.

MASCHINENMONITOR C16/C128

Ein Maschinenmonitor ist nicht so komfortabel. Die Adressen, die einmal geschrieben sind, sind geschrieben. Soll das Programm an eine andere Stelle kommen, so sind alle Adressen dementsprechend umzuschreiben. In unserem Falle wäre dies allerdings nur die JMP-Adresse. Lägen die Systemroutinen an anderer Stelle, so müßte man sie überall, wo ein solcher Aufruf erfolgt, abändern. Bei einem längeren Programm kann dies mit einigem Aufwand verbunden sein. Nun das Monitorprogramm:

```
. 3000 JSR  $FFE4
. 3003 JSR  $FFD2
. 3006 JMP  $3000
```

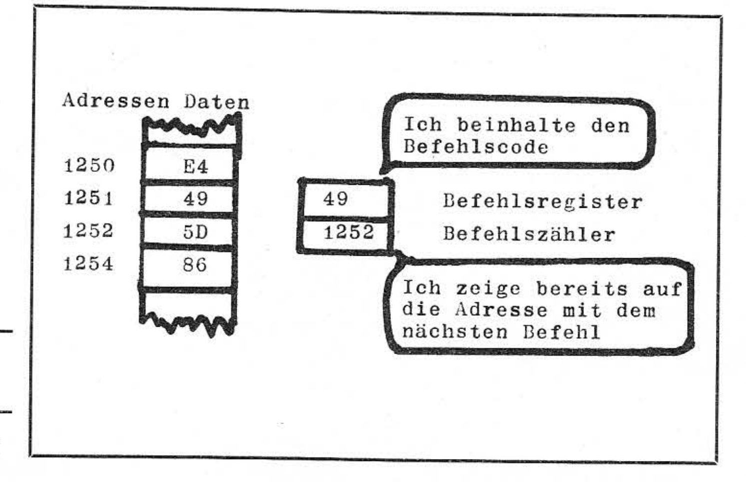
Links sieht man die Speicheradressen, rechts die richtigen Assemblermnemonics des 6502-Prozessors, gefolgt von den Adreßangaben. Auf die Hexzahlendarstellung haben wir verzichtet. Wir holen das aber schnell nach, wobei wir gleich die Dezimalzahlendarstellung beifügen.

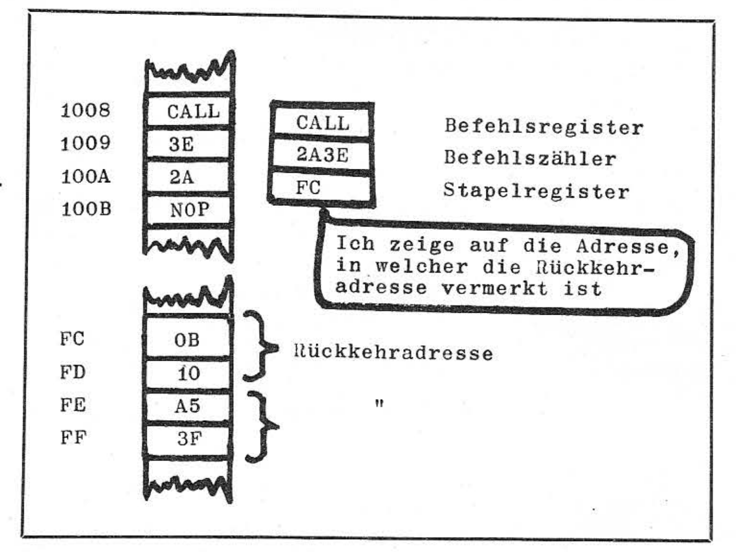
```
. 3000 20 E4 FF /
       032 228 255
. 3003 32 D2 FF /
       032 210 255
. 3006 4C 00 30 /
       076 000 048
```

Die Hexzahlendarstellung befindet sich normalerweise bei unserem Maschinensprachmonitor zwischen der Adreßangabe und den Mnemonics und wird vom Monitor selbsttätig erzeugt.
Ging beim Monitor gegenüber dem Assembler schon einiges an Lesbarkeit verloren, da man nicht vielsagende Begriffe vor sich hat, sondern sich Adressen merken muß, so tut man sich bei der reinen Zahlendarstellung besonders schwer, da man nun auch die Operationscodes im Kopfe haben muß. Ganz schlimm wird es aber, wenn die Daten als Datazeilen in Basicprogrammen vorliegen, da sich hier gar nicht mehr unterscheiden läßt, was nun Operationscode und was nun Daten oder Sprungadressen sind.

```
10 for i=1288 to 12296
20 reada:pokei,a:next
30 sys 12288
40 data 32,228,255,32,
   210,255,76,0,48
```

Dieses Basicprogramm lädt und startet nun das gewünschte Maschinenprogramm. Wenn Sie einen C16/116/Plus4 oder einen C64 oder 128 besitzen, so sollten Sie es ruhig einmal ausprobieren.





BEISPIEL: VERTEILER

In einem Programmzyklus erfolgt eine Messung. Als Ergebnis derselben, werden dem Computer 4 verschiedene Zustände mittels Flag0 und Flag1 signalisiert. Der Computer hat daraufhin eine von 4 verschiedenen Routinen auszuführen, wie in nachfolgender Tabelle dargestellt.

Flag1	Flag0	Routine
0	0	ROUTINE0
0	1	ROUTINE1
1	0	ROUTINE2
1	1	ROUTINE3

Nach der Ausführung der entsprechenden Routine soll wieder die nächste Messung erfolgen usw.

```
:BEGIN    CALL   MESSUNG
          BF1    VERT10
          BF0    VERT01
          CALL   ROUTINE0
          JMP    BEGIN
:VERT01   CALL   ROUTINE1
          JMP    BEGIN
:VERT10   BF0    VERT11
          CALL   ROUTINE2
          JMP    BEGIN
:VERT11   CALL   ROUTINE3
          JMP    BEGIN

:BEGIN    CALL   MESSUNG
          CALL   VERTEIL
          JMP    BEGIN
```

Übersichtlicher und kürzer, sowohl datenmäßig als auch zeitlich, ließ sich das Programm folgendermaßen schreiben:

```
:VERTEIL  BF1    VERT10
          BF0    VERT01
          JMP    ROUTINE1
:VERT01   JMP    ROUTINE2
:VERT10   BF0    VERT11
          JMP    ROUTINE3
          JMP    ROUTINE4
```

Da in der VERTEIL-Routine keine neue Rücksprungadresse auf den Stapel kommt, wird nach Ausführung der gewünschten Routine gleich wieder in die anfängliche Programmschleife zurückgesprungen.



BASIC, INTERPRETER, COMPILER, DATA-GENERATOR?

Eigentlich gibt es nur eine Sprache, die ein Computer versteht: Maschinencode. Genauer gesagt ist dies die Sprache, mit welcher der in einem Computer vorhandene Prozessor etwas anfangen kann.
Um einen Anwender aber den Umgang mit dem Computer zu erleichtern, stattet man ihn oft mit einem Hochsprachenteil aus. Im Falie der Commodore-Computer ist dies in vielen Fällen ein Basic-Interpreter, der fest in den Computer eingebaut ist.
Home-Computer vom Typ C64, C16/116 oder PLUS4 sind solche Computer.

INTERPRETER

Dadurch sind diese zwar befähigt, Basicprogramme abzuarbeiten, in Wirklichkeit spielt sich aber doch alles auf der Ebene des Prozessors ab und der Basic-Interpreter ist nur ein Hilfsprogramm (Dolmetscher), das bei einem Basicprogrammlauf kontinuierlich in die für den Prozessor verständliche Maschinensprache übersetzt. Diese Übersetzung erfolgt Satz für Satz (Befehlswort für Befehlswort mit den entsprechenden Parametern) und wird dann gleich ausgeführt.

COMPILER

Es gibt aber auch „Dolmetscher", die nicht simultan übersetzen, sondern einen vorliegenden Text (Programm) vor dem eigentlichen Programmlauf komplett übersetzen. In diesem Falle handelt es sich um Compiler.
Da aber auch Interpreter (bzw. deren Programm-Editoren) eingegebene „Texte" bei der Programmablage bereits aufbereiten, sind die Grenzen zwischen Interpreter


```
10 rem datagenerator ============16
20 rem (p)        commodore welt  ==
30 rem =============================
40 rem (c)        by              ==
50 rem            lothar miedel   ==
60 rem            commodore welt  ==
70 rem c 16/116/p4 u. cas/disk    ==
80 rem version 3.5  40z/ascii     ==
90 rem =============================
95 gosub 60000
110 :
120 rem definitionen und festlegung
en
130 rem a=datazeilenbeginn -1
140 rem az=anzahl der datas pro zei
le
150 dz=2000:az=5:dz=dz-1
160 :
170 print chr$(147)"datagenerator f
uer c16/116/plus4
180 print"--------------------------
-----------
190 rem oberen rand festlegen
200 poke 2022,3
210 :
220 rem cursor auf zeile 19 stellen
230 poke 205,19:print
240 print"--------------------------
-----------
250 :
260 input"file-name";f$
270 input"anfangsadresse (hexadezim
al) ";aa$:aa=dec(aa$)
280 input"endadresse.....(hexadezim
al) ";ea$:ea=dec(ea$)
290 :
300 rem fenster unterer rand - curs
or setzen
310 poke 2021,19:poke 205,5
320 :
330 print chr$(147)chr$(17)chr$(17)
chr$(17)"1000 rem "f$
340 print chr$(17)chr$(17)"1010 aa=
"aa":ea="ea":dz="dz+1:z=aa
350 :
360 print"dz="dz"+1:az="az":z="z":e
a="ea:rem datenuebergabe auf schirm
370 print chr$(17)chr$(17)"goto390"
:goto500
380 :
390 ps=0:printchr$(147):for j = 1 t
o 2:printchr$(17):next j:rem 390
400 print dz"data ";
410 for k=0 to az
420 if z<=ea+1 then d=peek(z):z=z+1
:ps=ps+d
430 if z>ea+1 then print"#"+str$(ps
```

```
-d):print:goto 510
440 print right$(hex$(d),2)",";:nex
t k
450 :
460 print "#"+str$(ps):print"dz="dz
"+1:az="az":z="z":ea="ea
470 if z>ea then 510
480 print chr$(17)chr$(17)"goto390"
490 :
500 print chr$(19):fori=1319to1324:
pokei,13:next:poke239,6:end:rem 500
510 poke 2022,0:poke 2021,24:print:
print:print"delete -999"
520 print chr$(19):for i=1319 to 13
29:pokei,13:next:poke239,10:end
1000 rem filename
1010 rem anfangsadresse - endadress
e - erste datenzeile
1020 print chr$(147)"adresse..strin
g..daten..peek..zeile
1030 poke 229,2:print
1040 for i= aa to ea+1:readd$
1050 if left$(d$,1)=chr$(35)then p$
=mid$(d$,2):p=val(p$):dz=dz+1
1060 if p then if s<>p then print"f
ehler in zeile: "dz-1:poke 2022,0:e
nd
1070 if p then s=0:i=i+1:if i>ea th
en print"fertig":poke 2022,0:end
1080 if p then p=0:s=0:i=i-2:next i
1090 if left$(d$,1)<> chr$(35) then
 d=dec(d$):s=s+d
1100 print i tab(9)d$tab(16)d;:poke
 i,d:print tab(23)peek(i)tab(29)dz:
next i
60000 rem nachspann ===============
60010 return
60020 rem ========================
60040 rem 002235 bytes program    ==
60090 rem =========================
```

und Compiler nicht ganz so scharf zu trennen. Trotzdem kann man davon ausgehen, daß dann, wenn ein Übersetzungslauf erforderlich ist, das Programm compiliert wird.
Ein menschlicher Dolmetscher kann nur dann richtig übersetzen, wenn ihm jedes der zu übersetzenden Wörter bzw. eine hierzu passende Übersetzung bekannt ist. Ist dies nicht der Fall, wird er versuchen, sinngemäß – aus dem Satz- und Thematikzusammenhang heraus – das unbekannte Wort zu ersetzen.
Ein Interpreter oder Compiler ist viel pingeliger! Kennt er einen Befehl nicht oder wird die Syntax bzw. die Anzahl der erforderlichen Parameter nicht eingehalten, so wird er durch "Syntax-Error" oder eine ähnliche Fehlermeldung darauf aufmerksam machen. Interpreter und Compiler kennen nur einen begrenzten Wortschatz, wodurch nicht alle in einem Computer eingebauten Möglichkeiten genutzt werden können. Computerbesitzer, die nur in Basic programmieren, können deshalb nur einen bestimmten Teil der Leistungsfähigkeit eines Computersystems ausnützen, da Basic außer den vorhandenen Basicbefehlen und der Beeinflussung durch PEEK und POKE keine weiteren Möglichkeiten bietet.

INTERPRETATION KOSTET ZEIT

Ein Nachteil eines Interpreters ist, daß für jede Interpretation Zeit benötigt wird. Ein Basicprogramm ist gegenüber den menschlichen Geschwindigkeiten zwar relativ schnell, aber für viele Aufgaben doch zu langsam. Reicht die Geschwindigkeit eines Basicprogrammes für eine Aufgabenstellung nicht mehr aus, dann kann entweder versucht werden, mittels einer anderen Hochsprache (z.B.: Forth, Pascal) das Problem zu lösen oder aber dem Prozessor alles sofort mundgerecht zu liefern, ihm das Programm also in Maschinensprache zu „servieren". Eine Möglichkeit hierzu sind die schon erwähnten Compiler, welche aus Hochsprachen-Programmen einen Zwischencode erzeugen oder aber die Basicbefehle direkt in Maschinensprache übersetzen. Es wird also nicht bei jedem Programmlauf interpretiert, sondern der Quellcode (das Basicprogramm) wird übersetzt und aus ihm entsteht ein ablauffähiges Programm, das dann gestartet wird. Da Basicbefehle aber aus mehreren, teilweise sehr vielen, einzelnen Maschinenroutinen bestehen, wird das übersetzte Programm meist länger als das ursprüngliche Hochsprachenprogramm. Man gewinnt dadurch dann zwar an Geschwindigkeit, verliert andererseits aber Speicherplatz. Meist wird dem überstzten Programm noch ein sogenanntes Run-Time-Modul verpaßt, das bei kurzen Basicprogrammen um einige Größenordnungen mehr Speicher benötigt, als das eigentliche Programm selbst. Bei umfangreichen Programmen kann es deshalb durchaus vorkommen, daß dem Compilat der Speicherplatz für den Programmlauf nicht mehr ausreicht. Dies trifft vor allem dann zu, wenn der Hochsprachenprogrammierer, der sein Programm compilieren will, die Besonderheiten des Compilers nicht kennt und ihm dadurch evtl. die Möglichkeit der Optimierung nimmt.

MASCHINENSPRACHE

Kommt es bei einem Programm vor allem auf Geschwindigkeit an, oder sind spezielle Effekte gewünscht, die mit einem eingebauten Interpreter nicht realisiert werden können, so kann Maschinensprache weiterhelfen. Dabei ist es gar nicht immer erforderlich, ein umfangreicheres Programm komplett in Maschinensprache zu schreiben.
In vielen Fällen reicht es, entsprechende Unterroutinen in dieser Sprache abzufassen. Außerdem hat diese Methode dann noch den Vorteil, daß diese Unterroutinen für verschiedene Programme immer wieder verwendet werden können, ohne diese neu schreiben zu müssen. Ich denke hier beispielsweise an Sortierprogramme, Hardcopy-Routinen usw.
Dies hat auch noch den Vorteil, daß z.B. Basicprogrammierer, die kein einziges Wort Maschinensprache beherrschen, auf derartige Routinen zurückgreifen können. Voraussetzung ist, daß sie in den Besitz dieser Routinen kommen.
Kleine, nützliche Maschinenprogramme werden oft in Zeitschriften abge-



druckt. Dabei gibt es drei „Druckarten".
Das **Assemblerlisting** ist vor allem für diejenigen interessant, die diese Routinen in eigene Programme einbauen wollen. Die Kommentare im Assemblerlisting helfen, das Programm zu verstehen. Derjenige, der Maschinensprache nicht beherrscht und das Programm nur einsetzen möchte, hat davon recht wenig und tut sich außerdem sehr schwer, das Programm einzugeben. Der kleinste Fehler und das Programm läuft nicht.
Eine andere Art der Veröffentlichung ist ein **Hexdump-Listing.** Dieser Art des Abdrucks ist außer den Speicherinhalten nichts zu entnehmen. Es ist nur eine Folge von

PLATZ GESPART MIT HEX-DUMP-LISTINGS

Speicheradressen und deren Inhalten, benötigt aber weit weniger Platz als ein kommentiertes Assemblerlisting. Auch hier gilt: Der kleinste Fehler und das Programm läuft nicht mehr korrekt. Die Fehlersuche ist relativ schwierig.
Deshalb werden in Zeitschriften oft **Ladeprogramme** (Hexloader, Dataloader, MC-Loader usw.) abgedruckt. Diese enthalten nichts anderes als die Speicherinhalte des Maschinenprogrammes, prüfen bei der Generierung aber, ob grobe Abtippfehler enthalten sind. Im einfachsten Falle wird überprüft, ob die Summe aller „Datazahlen" mit einer Prüfzahl übereinstimmt. Die Bemerkung am Ende eines Programmlaufes lautet dann oft: Fehler in den Datazeilen. Bei längeren Ladern kann die Suche nach dem Fehler dann sehr viel Zeit kosten.
Sinnvoll ist es deshalb, wenn am Ende einer jeden Datazeile sofort geprüft wird, ob in dieser ein Fehler enthalten ist. Auch dabei können nicht alle Fehler entdeckt werden, denn die Prüfsummen am Ende dieser Zeilen entsprechen ja der Addition der in der Zeile enthaltenen Datas. Vertauschungen von Daten werden dadurch nicht erkannt. Sicherlich könnten durch weiteren Programmaufwand auch noch die Summen der Spalten als Prüfzahlen Verwendung finden, aber hundertprozentig ist diese Lösung auch nicht! Man begnügt sich deshalb mit der einfacheren Lösung.
Es wäre für denjenigen, der Maschinenprogramme in dieser Form weitergeben will, aber sehr mühsam und fehlerträchtig, wenn diese Lader von Hand geschrieben werden würden. Deshalb benützt man zur Erzeugung „Datageneratoren". Das sind Programme, die aus Speicherinhalten Ladeprogramme generieren.
Ein Beispiel eines derartigen Generators für die Besitzer der Computer Commodore C16/116 und PLUS4 haben wir hier abgedruckt. Nach dem Programmlauf modifziert sich das Programm selbst, so daß nur noch das reine Ladeprogramm übrigbleibt. Die Zeilen zur Erzeugung des Ladeteils löschen sich selbst.
Bei diesem Programm wird außerdem gezeigt, wie man diese Computer sinnvoll überlisten kann, wie man die Fenstertechnik einsetzt und außerdem ist auf dem Bildschirm gut

DER GENERATOR IST DURCHSCHAUBAR

zu sehen, wie dieser Generator arbeitet. Wieviele Datas pro Zeile erzeugt werden, mit welcher Zeilennummer diese beginnen, kann im Programm verändert werden. Bei welcher Adresse begonnen werden soll und auch das Ende wird beim Programmlauf als Eingabe gefordert. Bei einem späteren Ladelauf kann gleichzeitig auch die korrekte Umsetzung verfolgt werden. *(LM)*

DIE BOOLSCHE WAHRHEIT

Sämtliche Commodore-Homecomputer (vom VC 20 bis zum PC 128) sind sogenannte 8-Bit-Rechner, d.h. in einer Speicherstelle (POKE) hat der Inhalt (PEEK) von 8 Bit = 1 Byte Platz.
Diese 8 Bit wiederum haben verschiedene Wertigkeiten (Bit 0 – Bit 7):

Bit	0 =	1 oder 0
	1 =	2
	2 =	4
	3 =	8
	4 =	16
	5 =	32
	6 =	64
	7 =	128

Als Mathematikgenie haben Sie natürlich sofort ausgerechnet, daß die höchste speicherbare Zahl 255 ist. Richtig, in einem Byte können Werte von 0 bis höchstens 255 stehen.
Diese Zahlen werden dann entsprechend Ihres Wertes als Buchstabe, Zahl, Grafik- oder Farbzeichen auf dem Ausgabegerät (Bildschirm, Floppy Datasette oder Drucker) ausgegeben. Nun gibt es aber noch eine andere Funktion von Speicherinhalten, die sog. Register. Hierbei soll das darin enthaltene Byte kein Zeichen darstellen, sondern jedes einzelne Bit (0–7) fungiert hier als Schalter, um eine ganz bestimmte Zustandsart des Computers zu bestimmen oder eine ganz bestimmte Tätigkeit ausführen zu lassen, je nachdem, ob dieses Bit nun eingeschaltet (=1) oder ausgeschaltet ist (=0).
Dies verhält sich zum Beispiel mit der Speicherstelle $D015 (53269) im Bereich des VIC-Chip (ab $D000) so: Sie regelt, ob einer oder mehrere Sprites (von 0–7) ein- oder ausgeschaltet sind. Wollen Sie zum Beispiel sicherstellen, daß Sprite Nr. 2 eingeschaltet ist, so müssen Sie die das 3. Bit (es hat -she.oben – die Wertigkeit 4) einschalten, ganz egal, was sich momentan im Speicherinhalt von $D015 gerade befindet:
POKE53269,PEEK (53269) OR 4.
Wollen Sie dieses nämliche Sprite wieder ausschalten, so sollten Sie von dem höchstmöglichen Speicherinhalt ausgehen (255). Die Anweisung lautet jetzt:
POKE53269,PEEK (53269) AND (255-4).
Nun haben Sie sicher schon bemerkt, worauf ich hinaus will:
Die AND (=UND) und die OR (=ODER)-Verknüpfung zweier Bytewerte.
Diese Verknüpfung dient zum Vergleich der gleichwertigen Bits eines Bytes. Man nennt dies die Boole'sche Wahrheitstabelle.

AND-Verknüpfung

Byte 1	11110001 = 241
Byte 2 (AND)	00001111 = 15
ergibt	00000001 = 1

Es wird im Vergleichsergebnis nur das Bit angeschaltet (auf 1 gesetzt) das im Byte 1 UND im Byte 2 auf '1' steht. Alles andere ergibt 'Null'. Prüfen Sie es mit Ihrem Computer nach:
print 241 and 15 ergibt 1 (und nicht 256, was rechnerisch ja logisch wäre)
Im Gegensatz dazu steht die OR-

OR-Verknüpfung

Byte 1	11110001 = 241
Byte 2 (OR)	00001111 = 15
ergibt	11111111 = 255

Hoppla, das sieht jetzt ganz anders aus! In der ODER-Verknüpfung steht immer dann im entsprechenden Bit eine '1', wenn diese '1' im selben Bit von Byte 1 ODER Byte 2 steht.
Eigentlich logisch, oder?

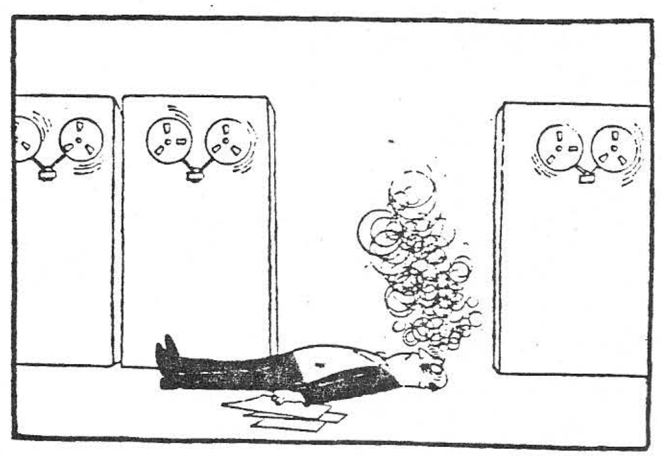
```
10 rem boolsche wahrheit=========16
20 rem (p)          commodore welt  ==
30 rem ============================
40 rem (c)          by              ==
50 rem        a. mittelmeyer        ==
60 rem        commodore welt        ==
70 rem                              ==
80 rem                              ==
90 rem ============================

95 gosub 60000
200 rem ***************************
210 rem farbwahl
220 rem     hintergrund: dunkles rot
230 rem     vordergrund: weiss
240 rem     rahmen      : dunkles gru
en
250 rem falls sie keinen c16 besitz
en
260 rem so passen sie folgene farb-
270 rem definition ihrem rechner an
,
280 rem oder lassen sie ganz weg
290 color4,6,3:color0,3,3:color1,2,
7
300 rem ***************************
***
310 gosub 320:goto 365
320 cl$=chr$(147)
330 print cl$:print:print
340 printtab(10)"boolsche wahrheit"
350 printtab(10)"================="
360 print:print:return
365 print"......werte bitte dual ei
ngeben"
370 print"......d.h. 8 ziffern, wob
ei nur"
380 print"......nullen und einsen e
rlaubt sind."
390 print:print:print
400 x$="1":gosub 520:a$=b$
410 x$="2":gosub 520
420 x$=a$:gosub 590:a=x
430 x$=b$:gosub 590:b=x
440 print:print"......o=or/a=and/x=
xor";:inputx$
450 if x$="o"then490
460 if x$="a"then500
470 if x$="x"then510
480 goto 440
490 y=aorb:fk$="or) ":goto670
500 y=aandb:fk$="and) ":goto670
510 y=(aandnotb)or(notaandb):fk$="x
or) ":goto670
520 print"......byte"x$": ";
530 input b$
540 if len(b$)<>8 then 520
550 i=1
560 if mid$(b$,i,1)<>"1"andmid$(b$,
i,1)<>"0"then 520
570 i=i+1:if i<9 then 560
580 return
590 x=0:for i=1 to 8
600 x=x+x+val(mid$(x$,i,1)):next
610 if x<0 then x=257-x
620 return
630 z=128:for i=1 to 8
640 if y<z then print "0";
650 if y>=zthenprint"1"; :y=y-z
660 z=z/2:next:return
665 printx$" ="x:return
670 gosub 320
680 print"....byte 1........";:x$=a
$:x=a:gosub 665
690 print:print"....byte 2 ("fk$"..
";: x$=b$:x=b:gosub 665
700 printtab(18)"--------"
710 print:print"....ergibt........"
;:x=y:gosub 630:print" ="x
720 print:print:print
60000 rem nachspann =============
60010 return
60020 rem =======================
60030 rem 012277 bytes memory    ==
60040 rem 001952 bytes program   ==
60050 rem 000000 bytes variables ==
60060 rem 000000 bytes arrays    ==
60070 rem 000000 bytes strings   ==
60080 rem 000000 bytes fre(0)    ==
60090 rem =======================
```

Auch hier können Sie es überprüfen, wenn Sie eintippen:
print 241 or 15 = 255
(aber wieder nicht 256!)
Als letzte interessante Boole'sche Wahrheit fungiert noch das ausschließliche ODER, genannt EXOR oder XOR.
Hierbei ist der Ergebniswert immer dann '0', wenn die beiden zu vergleichenden Bitwerte gleich sind (1 oder 0), bei ungleichen Werten ergibt sich '1'.

Byte 1	11110001 = 241
Byte 2 (XOR)	00001111 = 15
ergibt	11111110 = 254

Das Programm hierzu läuft ohne Spritedaten und Color-Anweisungen auf jedem Commodore-Computer



EINES FÜR ALLE

DAS UNIVERSELLE DRUCKER-INTERFACE

Die Wahl des Druckers fällt oft leicht, da jeder genaue Vorstellung zum Kauf mitbringt. Ein wenig schwerer wird's da schon bei der Auswahl des richtigen Interfaces und vor allem: bei der Programmierung dessen Funktionen, wenn's mal den Drucker mit dem Commodore-Computer verbindet.



Da bleibt gar nichts anderes, als das Druckerhandbuch und die Anleitung des soeben im Computershop erworbenen Interfaces zu studieren. Welche Sekundäradresse beispielsweise bewirkt Klein-/Großschrift auf dem Drucker? Welche Steuerzeichen sind nötig, um den grafikfähigen Drucker mit 24 Nadeln anstatt mit nur acht drucken zu lassen? (Voraussetzung ist natürlich, daß Ihr Drucker diese Möglichkeit vorgesehen hat.)

Nun, ab sofort könnte Ihnen ein hilfreicher "Alleskönner" zur Seite stehen, dem diese Probleme wie ein Kinderspiel erscheinen: Das "Printerface" von RTK, München, seit kurzem als eine überarbeitete Version des PCB C 128 Interface DZ 300 auf dem Markt.

Es wird über den seriellen Bus am Computer und mit dem Centronics-Stecker am Drucker angeschlossen, so wie's für ein Interface üblich ist.

GEEIGNET FÜR DIE GÄNGIGSTEN COMMODORE-HOMECOMPUTER

Die grundsätzlichen Pluspunkte wurden natürlich beibehalten:
Das "Printerface" eignet sich für den C 64 und den C 128 gleichermaßen, falls noch jemand den VC 20 + Drucker zu Hause

PROGRAMM-DISK FÜR DIE DRUCKER-EINSTELLUNG – MIT DIP-SCHALTERN FIXIEREN

stehen hat, wird er seine Texte oder Grafiken ebenso zu Papier bringen können.

Befinden Sie sich im "Commodore-Modus", so wird eine Druckausgabe analog zum Bildschirm erreicht, d.h. ein Klammeraffe ist auch auf dem Druckblatt ein Klammeraffe und kein Paragraphenzeichen oder Umlaut (die betroffenen Zeichen werden jedem bekannt sein!), die DIN-Tastatur des C 128 ist dem "Printerface" inzwischen so vertraut, als wären es die ASCII-Zeichen, ganz wichtig erscheint uns die Möglichkeit, daß auch Typenraddrucker angeschlossen werden können. Mit acht außen auf der Oberseite des Interfaces angebrachten DIP-Schaltern lassen sich die verschiedenen Optionen einstellen, wie Druckerart (Commodore-, Epson-kompatibel), Geräteadresse, ob ASCII- oder DIN-Zeichensatz, sowie die bereits erwähnten Sekundäradressen und letztendlich, ob der



Auto-Line-Feed aus- oder eingeschaltet sein soll.

Schieben Sie nun die dem Hardware-Paket beiliegende Diskette ins Laufwerk und laden Sie das Programm "Einstellung", nach 6 verschiedenen Testausdrucken unter Verwendung jeweils anderer Sekundäradressen werden Sie sehr schnell feststellen, welche für Sie in Frage kommen (nämlich die, die den Probeausdruck so exakt ermöglichen, daß es dem im Anleitungsbeiheft abgedruckten am nächsten kommt). Nun werden die erwähnten DIP-Schalter so eingestellt, wie das Programm "Einstellung" es Ihnen mitteilt. Mal ehrlich, geht's noch einfacher? Das ist nämlich schon eine der Neuerungen des verbesserten Interfaces, daß dieses Anpassungsprogramm vereinfacht und kaum noch "anfängerfeindlich" gestaltet wurde. Ein Wort an die C 128-Benutzer: es ist egal, in welchem Modus Sie das Installationsprogramm laden und starten, da es in reinem Basic 2.0 geschrieben wurde, läuft es in beiden Modi.

EXTERNE STROMVERSORGUNG

Normalerweise wird das "Printerface" nach dem Anschluß über Pin 18 der Centronics-Schnittstelle mit Strom versorgt, das ist jedoch von Drucker zu Drucker verschieden. Sollte das nicht funktionieren, so wird ein ebenfalls mitgeliefertes Kabel mit Verbindung zum Kassettenport Ihres Computers im Gehäuse des Interfaces angeschlossen (das Begleitheft gibt darüber genaue Auskunft). Das Interface bezieht jetzt seine Stromversorgung vom Kassettenport. Durch die vorhin erwähnte unkomplizierte Einstellmöglichkeit per Software hat kaum noch ein exotischer Drucker eine Chance, nicht mit dem "Printerface" zusammenzuarbeiten zu müssen, denn es simuliert praktisch alle gängigen und bekannten Interfaces, von Data Becker über Görlitz bis Merlin. Im übrigen ist auf der letzten Seite der Bedienungsanleitung eine Tabelle mit den Einstellpositionen der DIP-Schalter von 42 (!) verschiedenen Druckertypen abgedruckt. Falls Sie hier Ihren Drucker noch nicht gefunden haben, bleibt noch immer das Programm "Einstellung".

BEIM C 128 GEHT'S SCHNELLER –

Haben Sie das "Printerface" am C 128 angeschlossen, so wird der "Fast Serial"-Modus zur Datenübertragung genutzt und erreicht damit eine Geschwindigkeit von ca. 1300 Zeichen pro Sekunde. Optimal erweist sich in solch einem Fall die Möglichkeit der Zwischenspeicherung von Daten durch "Printerface".

Bei der 8-kByte-Version entspricht das etwa 3, bei der mit 32-kByte ca. 11 Seiten Text. Praktisch bedeutet das, daß beim Ausdruck eines vergleichsweise langen Schriftstückes keine Wartezeiten für den Computer entstehen. Es können andere Arbeiten erledigt werden, während der Ausdruck über das "Printerface" läuft.

DATENPUFFER IM INTERFACE

Übrigens ist dieser Puffer immer resident und

muß nicht eigens aufgerufen werden.

Unsere Tests ergaben, daß das Interface problemlos mit CP/M beim C 128 und "Geos" beim C 64 druckermäßig zusammenarbeitete. Vor dem Laden von "Geos" ist das ebenfalls auf der mitgelieferten Einstell-Diskette gespeicherte, kurze Ladeprogramm für "Geos" zu laden, das dem Drucker und der Floppy vor dem Nachladen der Benutzeroberfläche "Geos" wichtige Daten vermittelt.

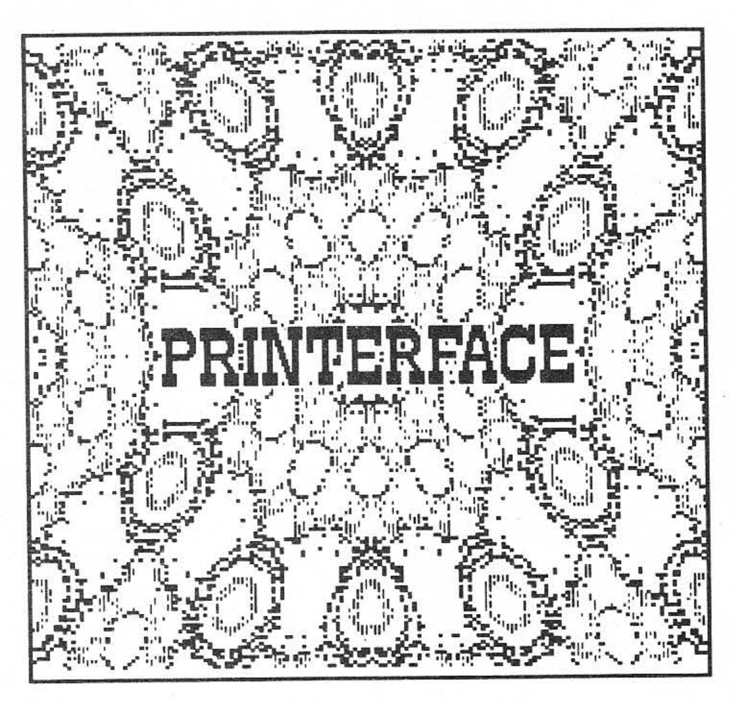
Im Gegensatz zu herkömmlichen Interfaces weist das "Printerface" keine festen Sekundäradressen irgendwelchen Funktionen zu. Es gibt praktisch vier Grundeinstellungen, Sie können aber die Sekundäradressen aller Funktionen frei wählen und bestimmen oder eine Funktion für alle Sekundäradressen festlegen.

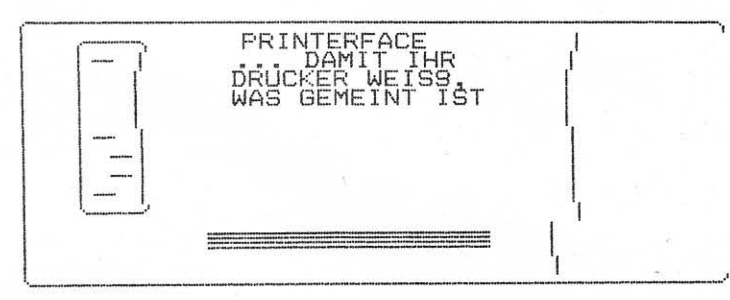
VIER GRUNDEINSTELLUNGEN

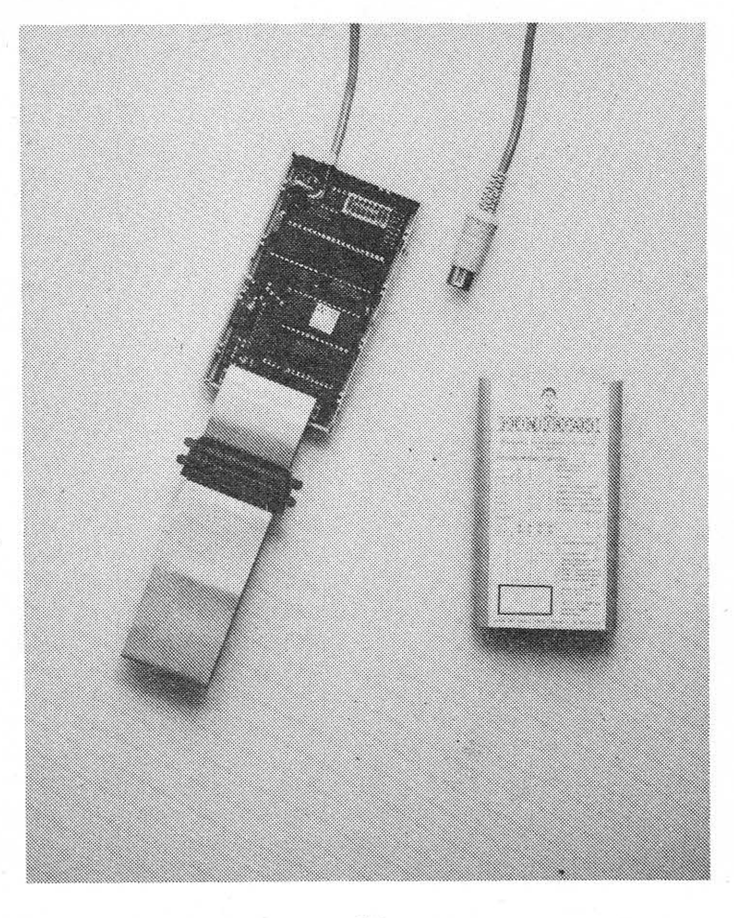
Folgende Grundeinstellungen sind vorgesehen: der Commodore (COM)-, Viza-, Data Becker und der Umlaut-Modus.

Sehr ausführlich ist in der Betriebsanleitung erläutert, welche Sekundäradresse für welche Funktion in Frage kommen. Kleine Beispielroutinen in Basic runden die Erklärungen über die einzelnen Modi ab.

Die Funktion Nr. 5 beispielsweise ist für denjenigen interessant, der seine mit einem Mal- oder Zeichenprogramm erstellten Bilder und Grafiken zu Papier bringen will.







Dieses Programm macht endlich Schluss mit lästigen Drucker-Problemen

Besitzt er nun einen (wie bereits eingangs erwähnt)

GESTOCHEN SCHARFER GRAFIK-AUSDRUCK

24-Nadeldrucker, so erreicht er aber erst dann einen gestochen scharfen und zufriedenstellenden Ausdruck, wenn er den Drucker auch mit 24-Nadeldaten versorgt. Üblich sind 8-Nadeldaten, was bedeutet, daß nur jede 3. Nadel angesprochen wird, und der geplante Ausdruck recht matt wirkt. Funktion Nr. 5 ermöglicht nun eine Umwandlung der Steuercodes für den Grafikdruck in 24-Nadeldaten, so daß jedes Byte der Grafik verdreifacht wird und so praktisch einen gestochen scharfen Schwarz-Weiß-Ausdruck ergibt.

FAZIT:

Dieses "Allround"-Interface macht nahezu Schluß mit lästigen Problemen, die sich aus der Nutzung von Textverarbeitungs- oder Grafiksoftware ergibt, falls sie nicht mit einem komfortablen Druckertreiber ausgestattet ist, bzw. Ihr Drucker darin nicht angegeben ist. Davon ausgehend, daß als letzte (und wichtigste) Neuerung das überarbeitete "Printerface" nunmehr für die 8 kB-Version um DM 50,– reduziert (DM 248,–), für die 32-kb-Version um DM 80,– billiger (DM 298,–) angeboten wird, können wir es aufgrund seiner Leistungsfähigkeit und der ausführlichen Bedienungsanleitung jedem interessierten Leser empfehlen, vor allem, wenn er sich beim Neukauf eines Druckers für ein Interface entscheiden soll. *(hb)*

KAMPF DEN ERRORS!

Selbst Profiprogrammierern mit langjähriger Computererfahrung unterlaufen Fehler. Zeitraubend ist die Suche nach den „Bugs", wie Errors in der Fachsprache genannt werden. Besonders Anfänger haben mit Errors Schwierigkeiten: Oftmals wissen sie nicht einmal, was die jeweilige Errormeldung des Computers bedeutet. Aus diesem Grund besprechen wir die einzelnen Fehlermeldungen des Basic 3.5 (Commodore C16) und führen auf, wie die Errors zu beheben sind.

Da C16, C64, C128 und PC10 ähnliche Basic-Interpreter haben, ist dieser Kurs auch für Benutzer der genannten Computersysteme wertvoll.

SPEZIELLES FÜR DEN C16–ANWENDER

Der Basic-Interpreter des C16 benutzt zur Fehlerangabe 2 Variablen: ER und EL.
In ER steht der letzte Fehler seit Start des Basic-Programmes, aus EL ist die Zeilennummer zu lesen, in der zuletzt ein Error auftrat. Die nachfolgenden Erklärungen gehen zum einen auf die ER-Nummer des C16, zum anderen auf den Ausgabetext ein, der durch die Befehlsfolge PRINT ERR$(ER) zu erhalten ist und den andere Basic-Interpreter bei Auftreten eines Fehlers ohne weitere Eingaben ausgeben.

ÜBER CODES, FEHLER UND TRICKS . . .

Errorcode des C16: 01
Text: Too Many Files Error

Öffnet ein Programm mehr als 10 Kanäle vom Rechner zu externen Geräten wie Laufwerk, Modem oder Drucker, gibt der C16 beim 11. Versuch, einen Kanal zu öffnen, diesen Fehler aus.
Gerne entsteht dieser Fehler, wenn man im OPEN-Befehl Variablen benutzt (Beispiel: OPEN A;B;C), die schlecht kontrolliert sind. Am besten prüft man die benutzten Variablen durch einen PRINT A,B,C-Befehl in einer neuen Zeile direkt vor dem OPEN-Befehl, um zu sehen, ob wirklich die geplanten Kanäle geöffnet werden.

Errorcode des C16: 02
Text: File Open Error

Soll der Basic-Interpreter einen Datenkanal öffnen, der bereits geöffnet ist, reagiert er mit diesem Fehler. Meist verursacht ein falscher Sprungbefehl, der die Abarbeitung des Programmes wieder zur OPEN-Anweisung führt, diesen Fehler.
Zweithäufigste Ursache ist, ähnlich wie unter Code 1, die unkontrollierte Nutzung von Variablen im OPEN-Befehl. Am Besten ist es hier, ebenfalls in einer neuen Zeile direkt vor den OPEN-Befehl einen PRINT A,B,C-Befehl zu stellen (s. Code 1).

Errorcode des C16: 03
Texte: File Not Open Error

Greift ein Programm auf externe Geräte wie Drucker, Floppy oder Bildschirm zu, ist zuvor ein Datenkanal hierfür zu öffnen. Gerne ist ein Tippfehler Ursache dieses Fehlers: Entweder in der OPEN-Anweisung oder in einer der folgenden GET#, INPUT# oder PRINT#-Anweisungen. Auch hier kann unkorrekter Inhalt einer benutzten Variablen Ursache sein (selten). Bei Verdacht auf diese Ursache prüft man vor jedem GET#, INPUT# und PRINT#-Befehl die benutzten Variablen auf deren Inhalt und vergleicht mit dem planmäßigen Inhalt.

Fehlercode des C16: 04
Texte: File Not Found Error

Ist am C16 eine Datasette angeschlossen und sucht der Rechenr auf der dort eingelegten Kassette nach Daten, gibt er einen File Not Found-Error am Bandende aus, wenn er die Bandende-Markierung liest. Floppy-Benutzer erhalten den Fehlercode 04 meist nach DLOAD-Befehlen, wenn entweder das einzulesende File falsch geschrieben wurde, die falsche Diskette eingelegt ist (auf der das gewünschte Programm nicht steht) oder der Diskettenhebel nicht verriegelt ist.

Errorcode des C16: 05
Text– Device Not Present Error

Öffnet man einen Datenkanal zu einem nicht angeschlossenen Peripheriegerät oder versucht man, ein File vom Laufwerk zu lesen oder zu schreiben, ohne daß die Floppy angeschlossen oder angeschaltet ist, erhält man diesen Fehlercode. Man prüft daher, ob alle Anschlüsse richtig sitzen und ob jedes Gerät Strom erhält.

Fehlercode des C16: 06
Text: Not Input File Error

Um aus einer Datei Daten zu lesen, muß man einen Datenkanal öffnen.
Der dazu verwendete OPEN-Befehl muß OPEN A,B,0 lauten. Das bedeutet: Variablen A und B sind frei wählbar, der dritte angegebene Wert muß 0 (für Lesen) sein. Meist handelt es sich bei diesem Error um einen Tippfehler.

Errorcode des C16: 07
Text: Not Output File Error

Die Erklärung dieses Fehlers ähnelt Fehlercode 06: Will man aus einer Floppy-Datei Daten lesen, muß der dritte Wert der OPEN-Anweisung 1 oder 2 sein, um dem Interpreter klar zu machen: Aus dieser Datei soll gelesen werden. Benutzt man den Wert 2 und Datasette, schreibt der Commodore automatisch nach Abspeichern aller Daten die Bandendemarkierung EOT auf Kassette.

Errorcode des C16: 08
Text: Missing File Name Error

Lädt man ein Programm von Diskette (DLOAD) oder transferiert Daten mit anderen Befehlen (DSAVE, R, W, . . .) zwischen Rechnern und Laufwerk, ist stets der Filename anzugeben. Fehlt der Name der Datei oder des Programmfiles, gibt der Basic-Interpreter den Missing File Name-Error aus.

Errorcode des C16: 09
Text: Illegal Device Number Error

Öffnet man einen Datenkanal auf den Bildschirm und versucht, von dort eine Datei auszulesen, gibt der Rechner diesen Fehler aus. Denselben Error erhalten Sie ebenfalls, wenn Daten vom Drucker eingelesen werden sollen. Derartige „Unmöglichkeiten" stammen meist aus Tippfehlern.

Errorcode des C16: 10
Text: Next Without For Error

In einer FOR-NEXT-Schleife (Beispiel: FOR I=1to50:PRINT I:NEXT) führt der Rechner sämtliche Befehle zwischen FOR und NEXT sooft aus, wie in einer Zählvariablen angegeben.
Stößt der Computer jedoch auf ein NEXT, ohne im FOR-NEXT-Modus zu sein, gibt er den NEXT-WITHOUT FOR-Error

aus. Meist liegt der Fehler darin, daß eine Variable verwechselt wurde. (Beispiel: FOR I=1to50: NEXT A)
Eine weitere Möglichkeit: Der Rechner wird durch einen GOTO- oder GOSUB-Befehl an eine Stelle des Programmes geschickt, an der er unvorbereitet auf ein NEXT stößt.

Errorcode des C16: 11
Text: Syntax Error
Häufigster aller Fehler. Jedesmal, wenn der Computer eine Anweisung absolut nicht versteht und keine nähere Erklärung für den Fehler finden kann, gibt er Syntax Error aus. Meist ist ein Tippfehler (GOWO, PLINT, etc.) der Grund. An zweiter Stelle steht das Benutzen von Basic-Schlüsselwörtern in Variablen. Die Anweisung: LET WORT$="Hallo" interpretiert der Computer als LET W OR T$ ="Hallo", da er OR als reservierten Befehl behandelt. Aus LET W OR T$ kann der Interpeter allerdings keinen Sinn erkennen: Deshalb der Fehler. Ein weiterer Grund ist oftmals vorheriges, sinnloses Herum-GePOKe im Betriebssystem. Dann hilft nur noch ein Hardware-RESET oder Ausschalten.

Errorcode des C16: 12
Text: Return Without Gosub Error
Ähnelt Fehlercode 10: Next Without For Error. Trifft der Interpreter auf den Befehl RETURN, ohne im GOSUB-Modus zu sein, gibt er den Error-Code 12 aus. Der Grund hierfür liegt in der GOSUB-RETURN-Anweisung: Springt der Interpreter in ein Unterprogramm, „merkt" er sich die Zeilennummer, von der er kam. Trifft er nun auf die Anweisung RETURN, nimmt er sich die „gemerkte" Zeilennummer und bearbeitet das Programm ab der folgenden Zeile weiter.
Ist allerdings keine Zeilennummer im Gedächtnis, gibt es auch keine Rücksprungzeilennummer: Also Error. Meist tritt der Fehler auf, wenn der Basic-Interpreter nach Abarbeiten des Hauptprogrammes ein folgendes Unterprogramm bearbeitet, anstatt auf eine END- oder STOP-Anweisung zu stoßen. Manchmal führt eine Verwechslung der Befehle GOSUB und GOTO zu dem Fehler. Bei GOTO gibt es kein RETURN.
Selten ist der Error eine Folge des Befehles POP. POP löscht die letzte Rücksprungszeilennummer im Gedächtnis des Rechners. Das Kurzprogramm

```
10  GOSUB 100
20  END
100 POP
110 RETURN
```

endet mit RETURN WITHOUT GOSUB ERROR in 110, da in Zeile 100 Befehl POP die gemerkte Zeilennummer 10 (siehe 10 GOSUB 100) löscht.

Errorcode des C16: 13
Text: Out Of Data Error
Eine beliebte Art, Maschinenspracheroutinen in Basic-Programme einzufügen, sind DATA-Anweisungen. Die Zahlenwerte des Unterprogrammes stehen in langen Kolonnen nach dem DATA-Befehl (Beispiel: 100 DATA 10,23,22,6) und werden durch eine FOR-NEXT-Schleife und READ ausgelesen, bevor der Interpreter sie per POKE in den Speicher schreibt (Beispiel:

```
10 FOR I=768to770
20 READ A: POKE I,A
30 NEXT I
90 DATA 1,2,3)
```

Manche Unterprogramme sind so lang, daß beim Abtippen der Kolonnen, ein Wert vergessen wird. Versucht das Programm in einem solchen Fall alle vorgesehenen Werte einzulesen, gibt es den OUT OF DATA ERROR. Denn die letzte einzulesende Zahl gibt es nicht. Damit ist auch gleich die Fehlerursache beschrieben. Beste Lösung ist, alle DATA's im eingetippten Programm auf Vollständigkeit zu prüfen. Seltenere Ursache ist, daß der Interpreter versehentlich zweimal in die READ-Schleife gerät, also ein zweites Mal versucht, alle Daten einzulesen. (Im obigen Beispiel wäre das mit der zusätzlichen Zeile: 40 GOTO 10 ermöglicht.) Um dieses Problem zu lösen, genügt in manchen Fällen der RESTORE-Befehl vor der FOR-READ-NEXT-Schleife.

Errorcode des C16: 14
Text: Illegal Quantity Error
Der C16 gibt Fehlercode 14 beispielsweise bei den Anweisungen PRINT A$(−1) oder POKE 257,0 aus. Der Grund: Matrizen nehmen immer positive Werte inklusive 0 an, in eine Speicherstelle des Computers passen maximal die Werte 0−255. Anweisungen mit Werten jenseits aller Möglichkeitsgrenzen führen zu ILLEGAL QUANTITY ERRORS. Meist sind diese Errors auf Tippfehler zurückzuführen, wobei der Tippfehler durchaus in einer Formel versteckt liegen kann. (Beispiel:

```
10 A=5
20 B=80
30 C=A*B/Fehler: Ursprünglich sollte hier C=A+B stehen
40 POKE 768,C)
```

Errorcode des C16: 15
Text: Overflow Error
Jeder Rechner hat Bearbeitungsgrenzen. Im Commodore 16 ist die größte Zahl 1.701411833E+38 und die kleinste −1.701411833E+37. Zahlen außerhalb dieses Bereiches führen zu OVERFLOW ERROR, egal ob die Zahl in einer Variablendefinition übergeben wird oder in einer Formel entsteht. Meist ist der Fehler Folge einer falsch eingegebenen Formel oder unkontrollierter Nutzung von Variablen in Formeln.

Errorcode des C16: 16
Text: Out Of Memory Error
C16/116-Nutzer ohne Speichererweiterung schlagen sich oft mit diesem Fehler herum. Im Programm selbst ist kein Fehler. Der Grund liegt im fehlenden Speicherplatz des Systems.
Gibt man ein Programm ein und füllt dabei den gesamten freien Basic-Speicher auf, kommt es bei der nächsten Programmzeile nach RETURN-Taste zum OUT OF MEMORY ERROR. In diesem Fall muß das Programm in 2 oder mehr Teilen eingetippt und auf Diskette gespeichert werden. Ein Programmteil lädt bei Bedarf weitere Folgeteile von Diskette nach. Weitere Gründe für Fehlercode 16 sind: Mehr als 10 FOR-NEXT-Schleifen sollten zur gleichen Zeit eröffnet werden. Meist liegt der Fehler in einem unkorrekt eingegebenen GOTO- oder GOSUB-Befehl, der den Interpreter immer und immer wieder an eine FOR-Programmstelle schickt. Dieselbe Zahlenbeschränkung ist bei DEF FN-Anweisungen zu beachten. Sind Programme bereits voll eingetippt und erscheint der Fehler erst nach Start des Programmes durch RUN, überlasten sehr wahrscheinlich Variablen den Speicher. Hier hilft die Funktion Variable =FRE(0). FRE(0) löscht alle nicht weiter benötigten Variablen und schafft somit Platz für neue Definitionen. Oftmals läuft ein Programm schon seit Minuten, bis der Error erscheint. Hier gilt dasselbe: FRE(0) muß her. Eine weitere Möglichkeit ist, von vorne herein möglichst wenige Variable zu verwenden und bei FOR-NEXT-Schleifen möglichst oft dieselbe Varia-

ble zu nutzen (I hat sich eingebürgert).

Fehlercode des C16: 17
Text: Undef'd Statement Error

Springt der Interpreter zu einer Zeile, die noch nicht eingegeben wurde, gibt es UNDEF'D STATEMENT ERROR.
Beispiel:

```
10 GOTO 100
20 END
```

führt zu UNDEF'D STATEMENT ERROR in Zeile 10.
Dasselbe gilt für die Basic-Anweisungen GOSUB Zeilennummer oder RUN Zeilennummer, falls Zeilennummer nicht vorhanden. Oft wollen Computer-Benutzer Programme starten, von denen sie gerade ein Drittel oder die Hälfte eingegeben haben. Tritt in solchen Fällen dieser Fehler auf, suchte der Interpreter eine Zeilennummer, die noch nicht eingegeben war. Bei eigenen Programmen sind meist Tippfehler entweder im GOTO/GOSUB/RUN-Befehl oder in der Zielzeilennummer die Ursache. Manchmal spielt das eigene Gedächtnis einen Streich, indem es eine Zeilennummer vorgaukelt, die nicht benutzt wird.

Errorcode des C16: 18
Text: Bad Subscript Error

Nutzen Programmierer Index-Variablen (Bsp. A(3), B(1,5)) ohne vorher alle Variablenfelder zu dimensionieren oder sind Matrizen zu klein dimensioniert, gibt der 16er diesen Fehler aus. Grundsätzlich gilt: Matrizen mit 11 Feldern (0 bis 10) müssen nicht DIMensioniert werden (DIM-Befehl-Beispiel: DIM A(20), B(10,15))
Weist man den Variablenmatrizen über 10 Werte zu, ist die entsprechende Variable vorher zu dimensionieren.
Beispiel:

```
10 DIM A(30): Für 30
   freie Variablen
20 FOR I=1 TO 30
30 A(I)=I
40 NEXT
```

Nach RUN erhält jede Variable den Wert ihrer Stellung. A(3) ist 3, A(23) ist 23.
Gibt man nun im Direktmodus:
LET A(31)=31 ein, gibt der Computer den BAD SUBSCRIPT ERROR aus. Oftmals entsteht der Fehler aus Folge einer Ausweitung der Programmkapazität. Im obigen Beispiel wäre das die Ausweitung der FOR-NEXT-Schleife auf über 30 Einheiten. Oft ist in „verbesserten" Versionen der ursprüngliche DIM-Befehl noch derselbe, was in der FOR-NEXT-Schleife zu Fehlern führt. Am Besten ist, Matrizen etwas größer als notwendig zu DIMensionieren. Voraussetzung hierfür ist genügend Speicherplatz, denn sonst gibt es bald einen OUT OF MEMORY ERROR!

Errorcode des C16: 19
Text: Redim'd Array Error

Manche Programmierer versuchen, Matrizen erneut zu DIMensionieren, weil die vorherige DIMensionierung zu gering war. Aus Variablenverwaltungsgründen geht das beim C16 nicht. Er gibt Fehlercode 19 aus. Es ist deshalb zu vermeiden, gleiche Variablenfelder mehrfach zu dimensionieren. Besser ist, von Anfang an größere Dimensionen anzustreben. Nicht genutzte Variablen einer Matrix benötigen sehr wenig Speicherplatz! Soll Auf Biegen und Brechen dasselbe Variablenfeld noch einmal DIMensioniert werden, muß vorher ein CLR durchgeführt werden. CLR löscht ALLE Variablen; sowohl Matrizen als auch String- und sonstige Variablen.

Errorcode des C16: 20
Text: Division By Zero Error

Beim Versuch, irgendeine Zahl durch 0 zu teilen, entsteht dieser Fehler. Selten liegt es daran, daß direkt eine Variable oder Zahl durch 0 geteilt wird. Viel öfter versucht der Interpreter, eine Variable durch eine Zahl oder Variable zu teilen:
C=B/A.
Hat A in diesem Fall den Wert 0, ist die Rechnung, so wollen es die Mathematiker, nicht durchführbar. Um diese Fehler zu vermeiden, prüft man die Variable A vor der Division auf ihren Wert und überspringt die Division, wenn A=0 ist. Beispiel hierfür:

```
10 IF A=0 THEN PRINT
   "A war 0!":GOTO 30
20 C=B/A
30 PRINT C
```

Errorcode des C16: 21
Text: Illegal Direct Error

Im C16-Basic gibt es zwei Modi: Den Direkteingabemodus und den Programmmodus. Der Direkteingabemodus ist zur Programmierung und direkten Kommunikation mit dem Betriebssystem des Rechners gedacht. Er benötigt keine Zeilennummer. Im Gegensatz zum Programmmodus, den der Computer dadurch erkennt, daß der Anwender vor der Befehlsfolge eine Zahl (Zeilennummer) schreibt. Einige Befehle funktionieren nur im Direkt-, andere nur im Programmmodus. Gibt man beispielsweise
INPUT "Eingabe"; A$
im Direktmodus, also ohne Zeilennummer ein, erscheint der ILLEGAL DIRECT ERROR. Dasselbe gilt für GET. Umgekehrt sind beispielsweise die Befehle DLOAD und DSAVE im Programmmodus nur in abgeänderter Form oder mit weiteren Funktionen versehen zu benutzen.

Errorcode des C16: 22
Text: Type Mismatch Error

Der C16-Interpreter unterscheidet zwischen 3 Variablenarten:
Stringvariablen wie A$ oder B$(3), Integervariablen wie A% oder B%(3), Zahlenvariablen wie A oder B(3).
In Stringvariablen stehen Buchstaben und/oder Zahlenkombinationen wie: "0012AS", "ttgr", "0000","Hallo".
In Integervariablen stehen geradzahlige Werte wie 12,10034 oder 0. Bei dem Befehl LET A% =12.234 behält die Variable A% nur die Werte bis zur Kommastelle, im obigen Beispiel den Wert 12. Zahlenvariablen nehmen alle Werte, ob geradzahlig oder mit Komma, an. In der Verarbeitung zwischen String- und Zahlen-/Integervariablen gibt es große Unterschiede: Zwar sind Strings genauso wie Zahlen miteinander addierbar, jedoch ist es unmöglich, Buchstaben einer Zahlenvariable zuzuweisen wie A="ABER" oder BALL="HER MIT".
Die Befehle RIGHT$, LEFT$, MID$ (siehe Handbuch) sind nur in Verbindung mit String-Variablen ohne Fehler anwendbar.
Um Zahlenvariablen in Stringvariablen umzuwandeln, gibt es den STR$-Befehl. Umgekehrt erhalten Anwender durch den VAL-Befehl alle Zahlen eines Strings bis zum ersten Buchstaben herausgefiltert. Meist ist Unachtsamkeit beziehungsweise Verwechslung zwischen einzelnen Variablentypen Ursache des Fehlers. Wichtigster Unterschied und weitverbreitetster Fehler sind Hochkommazeichen ("), die entweder vergessen oder unsinnig hinzugefügt wurden.

Errorcode des C16: 23
Text: String Too Long Error

Durch den 8 Bit-Prozessor bedingt, ist die Stringverwaltung des C16 auf bis zu maximal 255 Zeichen pro Stringvariable ausgelegt. Versucht man, einem String mehr als 255 Zeichen zuzuordnen, egal ob in einer LET-Anweisung oder durch Addition verschiedener Strings miteinander, gibt der C16 STRING TOO LONG ERROR aus. Derselbe Fehler tritt auf, wenn Programmierer versuchen, eine

Basic-Zeile mit mehr als 80 Zeichen (beim C16 entsprechend 2 Bildschirmzeilen) einzugeben. Sehr oft stehen Listings-Abtipper vor diesem Problem: Da stehen tatsächlich mehr als 80 Zeichen in einer Zeile. Der Grund: Im Commodore-Basic sind Befehle wie INPUT oder READ auch in Kurzform (Klein i groß N “iN“ für INPUT) einzugeben. Nach LIST zeigt der Computer die Befehle jedoch ausgeschrieben. Bei INPUT sind damit 3 Zeichen zu sparen. Aus diesem Grund sind bei überlangen Zeilen in Programmlistings stets alle Basic-Befehle in Kurzform einzugeben. Eine Tabelle mit Kurzschreibweisen ist dem Handbuch angegliedert. Letzte Ursache für diesen Fehler: Gibt der Anwender bei Abfragen durch INPUT mehr als 80 Zeichen ein, erscheint ebenfalls Fehlercode 23.

Errorcode des C16: 24
Text: File Data Error

Dieser Fehler entsteht, wenn falsch abgespeicherte Daten von Kassette wieder in den Computer geladen werden sollen. Sind Daten beispielsweise nicht korrekt durch CHR$(13) entsprechend RETURN-Taste getrennt, kommt es zum Überladen des Stringspeichers nach 255 Zeichen. Dasselbe gilt beim Einlesen von Daten von Diskettendateien.



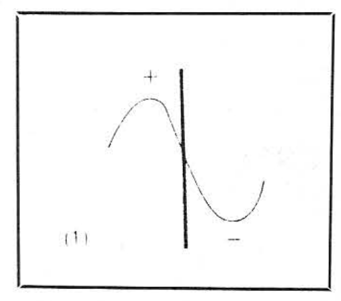
LOAD ERROR

Wenn man sich einen Computer gekauft hat, stellt man bald fest, wie unangenehm es ist, daß der Computer durch das Ausschalten alles vergißt, was man ihm mit viel Mühe beigebracht hat. Man braucht also eine Möglichkeit, um die eigene Arbeit sicher zu lagern, wenn der Computer seine Ruhepause macht. Da alles, was der Computer weiß, aus Nullen und Einsen besteht, liegt es nahe, diese Informationen magnetisch aufzuzeichnen, zum Beispiel auf einer Kassette.

Zu einem Preis von ca. 80,– DM bietet Commodore ein Kassettenlaufwerk an, das es ermöglicht, die eingegebenen Daten auf Band zu speichern und auch wieder in den Computer einzulesen. Und hier kommt nun das Problem.
Wer mit seiner Datasette Programme eines anderen Kassettenanwenders laden will, der erlebt nach Eingabe des LOAD-Befehls oft die sonderbarsten Dinge. Neben einem lapidaren LOAD-ERROR kann er aber auch Meldungen wie ‘FILE NOT FOUND, FOUND‘)<>? !$%$!§‘ OUT OF MEMORY oder ähnliches erhalten. Im schlimmsten Falle befindet sich der Computer plötzlich im Maschinensprachemonitor, oder er stürzt einfach ab. Dieses alles bedeutet nicht, daß der Computer defekt ist, sondern, daß die Datasette auf Grund eines verstellten Tonkopfes nicht in der Lage ist, die Aufzeichnung so zu lesen, wie es für eine ordnungsgemäße Nutzung nötig wäre. Wobei es natürlich nicht an dem Tonkopf Ihrer Datasette liegen muß, denn auch der eventuell verstellte Tonkopf Ihres Tauschpartners trägt dazu bei, daß allmählich Frust bei den Kassettenbenutzern aufkommt.

DIE URSACHE

Um Tonaufzeichnungen jeglicher Art auf magnetische Tonträger (Bänder, Kassetten, Disketten) optimal zu speichern bzw. wieder zu laden, müssen die Tonköpfe eine genau definierte Lage zu dem Speichermedium einnehmen. Auf gut deutsch, bei der Kassette muß der Kopfspalt genau senkrecht zum Band stehen. Der Fachmann nennt dieses den AZIMUT.
Die Justierung des Wiedergabe-Kopfes ist sehr wichtig, denn es besteht ein direkter Zusammenhang mit der Qualität



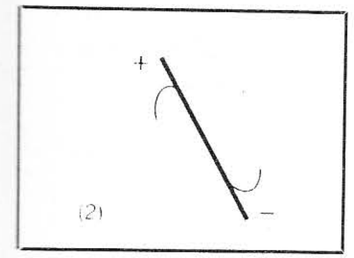
Korrekt eingestellter Tonkopf, der durch seine genaue Justierung in der Lage ist, die Signale auf dem Band richtig zu lesen und die magnetischen Aufzeichnungen getreu wiederzugeben.

der wiedergegebenen Signale. Die Massenproduktion der Bänder und der verstärkte Einsatz von Turboladern, aber auch die auf dem Markt erhältlichen Computersets (C16/116/Plus4), die leider immer noch (Preisvorteil) mit Datasetten ausgeliefert werden, hat diesem Problem immer mehr Bedeutung



Commodore Datasette 1530 (VC-20/C-64). Außer der Farbe und dem Anschlußstecker mit der 1531 für den C16 identisch.

gebracht. Die meisten Bänder, die wegen eines LOAD-ERRORS zurückgegeben werden, sind nicht auf schlecht hergestellte Bänder, sondern auf einen falsch justierten Tonkopf zurückzuführen.
Folgende Lösungen bieten sich nun an: Sollte der Computer etwas auf

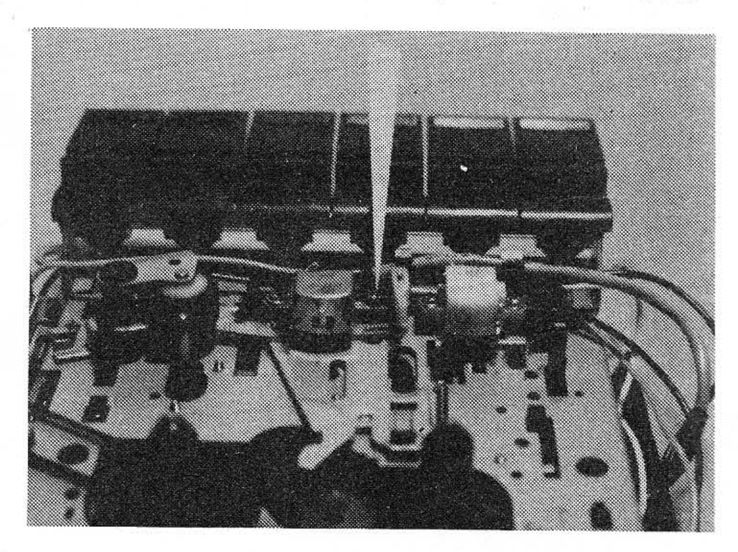


Falsch eingestellter Tonkopf (übertrieben). Der Spalt des Kopfes steht diagonal zum aufgezeichneten Signal. Dadurch liest der Kopf gleichzeitig 'Plus' und 'Minus'-Signale. Dies hat ein Auslöschen der Signale bei hohen und Verzerrungen bei niedrigen Frequenzen zur Folge.

dem Band finden, dann läßt sich die Dejustierung in drei Klassen einteilen:
Der Computer meldet sich mit FOUND Filename, lädt einige Sekunden bis Minuten und meldet dann einen LOAD ERROR:
Der Tonkopf ist nur leicht verstellt oder die Kassette ist defekt, was aber in den meisten Fällen auszuschließen ist.
Der Computer meldet sich mit FOUND Filename und anschließend sofort mit LOAD ERROR oder OUT OF MEMORY ERROR: Hier liegt eine größere Verstellung vor.
Der Computer meldet sich mit FOUND irgendwas, wobei irgendwas sich aus Grafikzeichen und allen möglichen Zeichen zusammensetzt: Da hapert es gewaltig.
Findet der Computer aber gar nichts und die Datasette schaltet am Bandende aus, ist guter Rat teuer. Geschehen bei einer von drei neuen Redaktions-Datasetten.
Hier bewahrheitete sich meine Vermutung wieder einmal, die da lautet, ‚Deckel auf, Tonkopf reinfallen lassen, einpacken, verkaufen'.
Was war nun der Fehler, werden Sie sich fragen. Ich wage gar nicht, dies mitzuteilen. Wie aus der Zeichnung zu ersehen ist, wird der Tonkopf, der auf einem Metallstreifen aufgeschweißt ist, mittels zweier Schrauben im Recorder befestigt, wobei eine Schraube fest angezogen ist und die andere mittels untergelegter Feder für die Tonkopfeinstellung zuständig ist. Meistens sind die Schrauben mit einem Sicherungslack versehen, die die Verstellung durch mechanische Bewegungen verhindern soll.
In unserem Falle war dieses auch der Fall, nur, daß die feste Schraube überhaupt nicht fest war, sondern sich nach ca. 4 Umdrehungen festdrehen ließ. Dadurch war natürlich keine genaue Position des Tonkopfes möglich, woran auch unsere ersten Einstellversuche scheiterten. Sollten Sie also zu keinem Einstellergebnis kommen, kontrollieren Sie einmal die andere Schraube am Tonkopf.

EINSTELLEN

Legen Sie nun die Kassette mit dem zu ladenden Programm ein. Verdrehen Sie die Einstellschraube um ca. ein bis zwei ‚Stunden' in eine beliebige Richtung. Merken



Tonkopf mit Verstellschraube

> **Anschriften:**
> *CSJ-Soft,*
> *An der Tiefenriede 27,*
> *3000 Hannover 1*
>
> *Aackosoft International,*
> *PO Box 3111,*
> *2301 DC Leiden/*
> *Niederlande*
>
> *Robcom,*
> *Mastertronic GmbH,*
> *Kaiser Otto Weg 18,*
> *D-4770 Soest*

Sie sich die Richtung, in die Sie die Schraube gedreht haben. Machen Sie nun einen Ladeversuch. Jetzt gibt es wieder verschiedene Möglichkeiten. Verändert sich überhaupt nichts, drehen Sie die Schraube um den entsprechenden Stundenwert in die andere Richtung und machen einen neuen Ladeversuch. Ist der Fehler kleiner geworden, drehen Sie die Schraube um eine weiteren Stunden-

AUFS NEUE BITTE

wert in die gleiche Richtung. Anschließend erfolgt ein neuer Ladeversuch, usw. usw.
Irgendwann fällt einem dieses natürlich auf den Wecker und es wird nach einer anderen Lösung gesucht.
Solange Sie nur Ihre eigenen Programme verwenden, bzw. käufliche Software einsetzen, sollten Sie danach trachten, Ihren Recorder optimal einzustellen. Hierzu bieten verschiedene Firmen entsprechende Programme an, die ein Einstellen des Tonkopfes erleichtern. Meistens bestehen die Programme aus einem Maschinenprogramm, das eine Frequenz auswertet, die sich auf der gleichen Kassette befindet. Genaueres entnehmen Sie bitte den entsprechenden Bedienungsanleitungen.
Aber auch Ihr Commodore-Händler kann Ihnen bei diesem Problem helfen. Für einen gewissen Obolus wird er Ihnen die Datasette präzise einstellen.

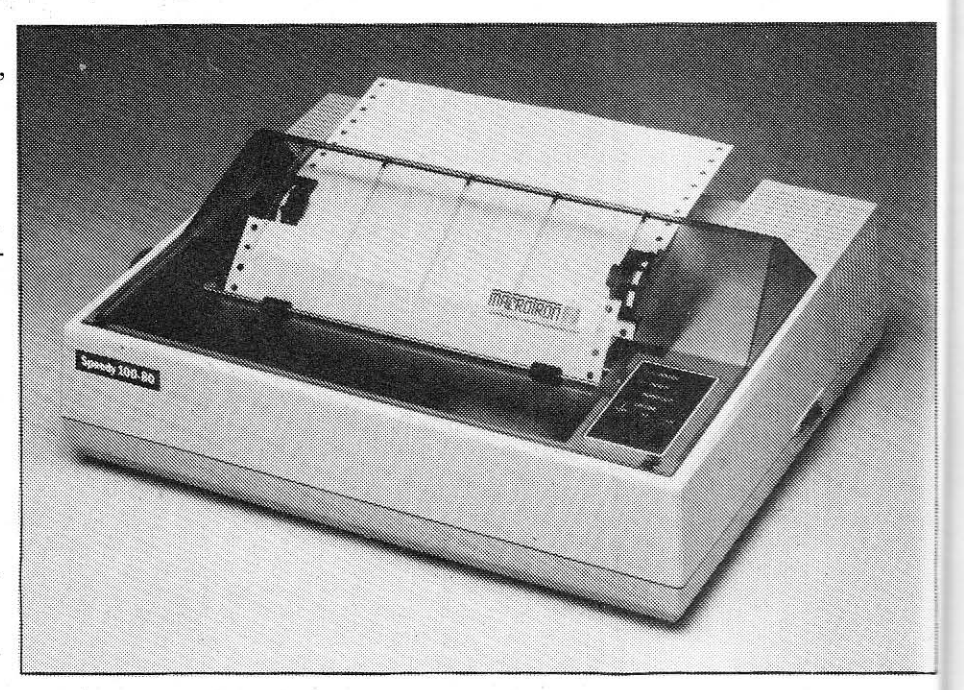
B.W.

LÜCKE GESCHLOSSEN: TEXTVERARBEITUNG AUCH FÜR C16!

Der private Computeranwender, der des Spielens überdrüssig geworden ist, sieht sich über kurz oder lang nach anderen Aufgaben um, die sein Computer übernehmen kann. Zwangsläufig zieht er außer Dateien auch eine Textverarbeitung in Betracht. Für die meisten Homecomputer, wie C-64 und 128 PC, gibt es diese Programme schon in sehr großer Zahl, so daß man hier schon fast die Übersicht verliert. Anders sieht es bei den "Bonsai-Computern" (C-16/C116) von Commodore aus.
Eine Ausnahme bietet die doch etwas magere Textverarbeitung des Plus 4. Für den C-16/C116 gibt es, soweit bekannt, nur eine Textverarbeitung der Firma Kingsoft.

STECKMODUL SCHAFFT SPEICHERPLATZ

Die, da sie in den Speicher geladen wird, läßt nicht mehr viel Speicherplatz für den Text übrig. Diese Lücke schließt nun das neue Textverarbeitungssystem "Script/Plus" von Commodore. Als Modul für den Expansionsport belegt es keinen Speicherplatz und kann sich doch mit vielen hochwertigen Textverarbeitungen messen, die für andere Computer angeboten werden.
Nach dem Einschalten und Starten des Moduls mit der F2-Taste ist derjenige, der das Programm Easy-Script kennt, doch etwas überrascht, denn das Titelbild ist zu 90% mit diesem identisch. Kein Wunder, kommt es doch aus dem gleichen Softwarehaus.
Nach dem Einschalten und vor Eingabe des Textes muß der Benutzer nun noch einige Grunddaten eingeben. Die Anzahl der Zeichen pro Zeile geht von 40 – 240 Zeichen pro Zeile.



Ein preisgünstiger Drucker für den C16/Plus 4: der Speedy

Man sollte die angegebene Zeichenzahl (40) beibehalten, da sonst der Bildschirm seitlich auswandert und man dann leicht die Übersicht verliert. Die höhere Zeichenanzahl sollte man bei der Erstellung von Tabellen anwenden, da man hier sowieso mit der Tabfunktion arbeitet, um die einzelnen Punkte wiederzufinden.
Als nächstes wählt man zwischen Disketten- und Kassettenspeicherung aus. Auch die Druckerauswahl wird in diesem Menüpunkt vorgenommen. Hier stehen schon Drucker zur Verfügung. Nach Eingabe aller Daten gelangt man in den Editier-Modus. Jetzt kann man den gewünschten Text eingeben. Über das Zeilenende oder ähnliches braucht man sich keine Gedanken zu machen, denn das ist Aufgabe des Formatierens. Mit Hilfe des Trennungsvorschlages, der unter Nutzung der Escape und Commodore-Taste erfolgt, kann das Programm dann gegebenenfalls die Wörter trennen. (Anmerkung: Auf Seite 75 Abs. 3,11 des Handbuches steht es etwas anders!)
Hat man den Text vollständig eingegeben, kann man daran gehen, den Text zu formatieren. Dazu stehen dem Benutzer eine Vielzahl von Befehlen zur Verfügung. Diese werden immer an die Stelle im Text gesetzt, ab der sie wirksam werden sollen und beeinflussen den Ausdruck, bis sie entweder aufgehoben oder durch neue ersetzt werden. Vor jeder Formatierungseingabe muß die Tastenkombination >CONTROL 9< benutzt werden.

UMFANGREICHER BEFEHLSSATZ

Folgende Befehle stehen zur Verfügung:
- rechten Rand setzen
- linken Rand setzen
- Papierlänge
- Textlänge
- Zeilenabstand
- Zentrieren
- Blocksatz
- rechtsbündiger Rand
- Zeilenvorschub

- Leerzeichen
- Rand lösen
- erzwungener Seitenumbruch
- Kommentarzeilen
- Schreibdichte
- Zeilendichte
- Seitenanfang
- Hoch- und Tiefschreibung
- Unterbrechung

und noch viele mehr.

Außer der vielfältigen Druckerausgabe können alle Texte natürlich auch auf Kassette oder Diskette abgespeichert und wieder geladen werden.

Als Besonderheit kann man hier betrachten, daß Text-File des Script/Plus und des Easy/Script untereinander ausgetauscht und auch bearbeitet werden, wenn man die unterschiedlichen Formatbefehle in Betracht zieht. Das trifft besonders auf den inversen Stern zu, der mit dem entsprechendem Programmbefehl geändert werden muß.

Eine weitere Besonderheit ist der eingebaute Rechner, der alle Grundrechenarten incl. Prozentrechnen im Text oder extern ermöglicht. Dadurch ist natürlich das Erstellen von Rechnungen und dergleichen zu einem Kinderspiel geworden.

Für Plus 4-Besitzer, die die etwas magere eingebaute Textverarbeitung leid sind, bietet Script/Plus etwas ganz Besonderes. Durch seinen großen Speicher bedingt, können zwei verschiedene Texte gleichzeitig im Speicher stehen und auch bearbeitet werden. Hierzu bietet das Programm den SWAP-Befehl an, der ein Umschalten zwischen den Texten ermöglicht.

ZWEI TEXTE GLEICHZEITIG BEARBEITEN

Der wichtigste Teil eines Textverarbeitungsprogrammes ist natürlich die Be- und Verarbeitung von Texten. Auch in diesem Punkt hat Script/Plus einiges zu bieten:

- Löschen von größeren Textmengen
- Löschen von Buchstaben, Zeilen und Textblöcken
- Einfügen
- Verschieben
- Suchen
- Suchen und Austauschen
- Überspeichern
- Einbau einer Kopf- und/oder Fußzeile
- Unterstreichen und Fettdruck
- Form und Serienbriefe
- Tabulatoren und Zahlenbetrieb
- Globale Operationen:
  * Suchen
  * Suchen und Austauschen
  * Ausgabe auf dem Bildschirm
  * Ausdrucken

Script/Plus bietet dem Benutzer auch noch die verschiedensten Lösch- und Kopierfunktionen.

Auch können Textteile, die auf dem Drucker vergrößert dargestellt werden sollen, auf dem Bildschirm revers abgebildet werden. Such- und Austauschfunktionen sind natürlich auch in diesem Programm enthalten. Diese beziehen sich auf den Text im Hauptspeicher wie auch auf die Texte, die auf der Diskette gespeichert sind. Da wir schon einmal bei der Diskettenstation sind: Hier stehen dem Anwender alle Befehle der Floppy zur Verfügung. Das reicht vom Formatieren bis zum Einlesen und Ausdrucken der Directory.

Texte, die durch ihre Länge nicht mehr in den Speicher passen, können durch Verkettungsbefehle auf dem Drucker ausgegeben werden.

Auch besteht die Möglichkeit, mittels Datafiles Serienbriefe zu erstellen, wobei die Daten über spezielle Befehle an die richtige Stelle des Textes gesetzt werden.

MIT SCRIPT/PLUS BEKOMMT DER DRUCKER FUTTER

Da Script/Plus als Modul im Expansionsport betrieben wird, kann es durch einen Befehl ein- und ausgeschaltet werden. Auch kann während des Arbeitens mit dem Programm ein Neustart durchgeführt werden, um z.B. die Zeichenzahl oder den Drucker zu wechseln, ohne daß der Text im Speicher verlorengeht.

Nach so viel Lob hält das Programm auch einen Tadel aus: das Handbuch. Nicht, daß es schlecht wäre. Nein. Aber, obwohl es mit ca. 150 Seiten die Fähigkeiten des Programms sehr gut und ausführlich erklärt, hat es doch einen schwerwiegenden Fehler, es ist nur in englisch erhältlich, was den deutschsprachigen Benutzer doch vor einige Probleme stellt.

FAZIT

Das wohl beste Textverarbeitungsprogramm, das für die kleinen Commodore-Computer zur Verfügung steht. Auch Plus 4-Anwender sollten sich dieses Programm einmal ansehen, da es das eingebaute um Längen schlägt.

Preis: 39,– DM

Bernd Welte



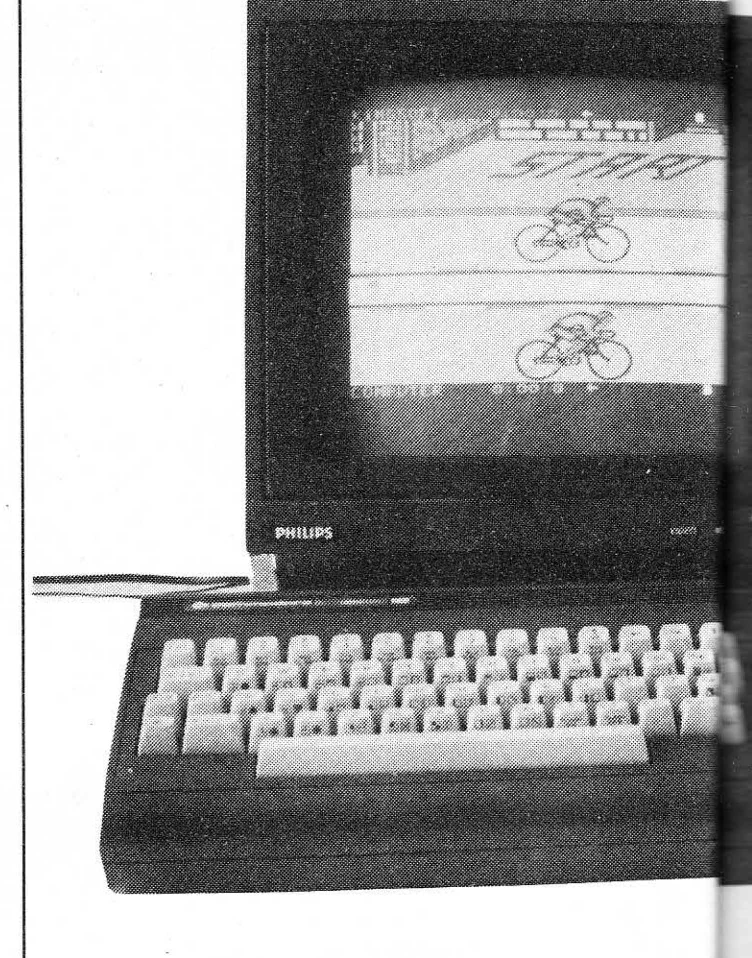




LERNEN SIE IHREN COMPUTER KENNEN

DIE COMMODORE-COMPUTER C16, C116 UND PLUS/4 BESSER PROGRAMMIERT.

Aufgrund der äußerst günstigen Preise leisten sich gerade Computereinsteiger die „kleinen" Commodores mit den Typenbezeichnungen C16, C116 und Plus4. Für den Kaufpreis wird auch einiges geboten, das sich wahrlich sehen lassen kann. Vor allem die Basicversion 3.5 zeigt sich dem wahrhaft spartanischen V2, mit welchem der „Commodore-Renner" C-64 ausgestattet ist, weit überlegen.



Starkes Basic: Commodores „Kleine"

Die Typen C16, C116 und Plus4 haben zwar gegenüber dem C-64 einen wesentlich verbesserten Basic-Interpreter, aber das muß nicht immer von Vorteil sein. Denn ein leistungsstarkes Basic zeichnet sich nicht zuletzt auch dadurch aus, daß der Befehlsvorrat sehr umfangreich ist. Und genau da „hakt" es dann bei völligen Computerneulingen, die Schwierigkeiten haben, überhaupt einmal einen Überblick über die Basicbefehle zu bekommen. Mit anderen Worten, diese Befehle müssen erst einmal erlernt werden.
Die wenigen im Handbuch enthaltenen Beispiele reichen einfach nicht aus, um wirklich Basic zu lernen. Hierzu müssen eben weitere eigene Programme geschrieben werden und auch die Analyse von fremden Programmen ist ein guter Weg, das Wissen und Können zu verbessern.
Gleiches gilt natürlich nicht nur für die Programmiersprache BASIC, sondern überhaupt für den gesamten Umgang mit dem Computer.

ABSCHAUEN GILT

Viele Tricks und Kniffe lernt man durch „spicken", nämlich durch zusehen, wie andere arbeiten. Aber natürlich auch durch uns, denn wir bringen immer wieder viele Programme, viele Informationen und jede Menge an nützlichen kleinen und großen Hilfen für den täglichen Umgang mit dem Computer. Das Lesen dieser Zeitschrift ist also immer ein Gewinn, selbst wenn das eine oder andere Programm mal nicht in Ihr Interessengebiet fällt.
Gerade für Leute, die mehr über ihren Computer wissen wollen, aber auch für diejenigen, die erst einsteigen bzw. einsteigen wollen, haben wir uns nun etwas Besonderes ausgedacht: Wir bringen eine ganze Serie von Artikeln, die zeigen, wie Sie nicht nur lernen bessere Programme zu schreiben, sondern Ihren Computer wirklich beherrschen können und nicht umgekehrt. Diese Artikelserie, die ab sofort in jedem Heft der COMMODORE WELT zu finden ist, bietet für jeden, der einen der bereits genannten Computer besitzt, etwas. Also, lassen Sie sich nicht lange auffordern, machen Sie mit und Sie werden sehen, in sehr kurzer Zeit kennen und beherrschen Sie Ihren Computer besser als manch einer, der Ihnen heute noch als Computerfreak erscheint. Dabei müssen nur zwei Voraussetzungen erfüllt sein, um erfolgreicher mit einem der genannten Computer umzugehen. Die eine ist, daß Sie überhaupt Zugang zu einem der aufgeführten Computer haben und die andere ist Ihr Wille, mitarbeiten zu wollen. Alles andere bringen wir Ihnen bei. Einverstanden?
Wie Sie sicher auch schon selbst wissen, kann man am meisten lernen, wenn man praktisch mitarbeitet. Deshalb werden immer viele kleine Listings zu finden sein, die Ihnen Ihren Computer näherbringen. Diese sollten Sie abtippen, um all das von uns Geschriebene immer gleich nachvollziehen und noch weiter vertiefen zu können.
Also nochmals: Durch lesen, anhören und zusehen lernt man zwar sehr viel, aber der größte Lernerfolg wird immer noch dann erreicht, wenn



man selbst arbeitet bzw. mitarbeitet. Ansonsten bleibt alles graue Theorie! Zwar wird es anfangs für viele sehr leicht und problemlos aussehen, was wir Ihnen beibringen wollen, aber wir können nur raten: Bleiben Sie am Ball, denn sonst verlieren Sie ganz schnell den Faden und müssen dann doch erst wieder alles nachlesen.

DIE COMPUTER

Diese Artikelserie beginnt zwar auf Einsteigerniveau, aber wir steigern uns dann von Mal zu Mal, so daß auch Fortgeschrittene sehr bald eine Menge lernen können. Machen Sie mit, arbeiten Sie die Beispiel-Programme durch, dann werden Sie noch viel mehr Freude an Ihrem elektronischen Partner haben.
Es geht nun nicht nur darum, welche äußerlichen Merkmale die einzelnen Rechner unterscheiden, sondern vor allem um den prinzipiellen Aufbau, den Sie nun erst einmal kennenlernen sollen. Keine Angst, wir bilden Sie dabei nicht zum Elektronik- oder Computertechniker aus, sondern werden versuchen, Ihnen alles möglichst einfach und verständlich nahezubringen. Wir berücksichtigen dabei auch, daß aufgrund der unterschiedlichen Altersstruktur der Besitzer dieser „Einsteigermodelle", in vielen Fällen nur wenig Vorkenntnisse vorliegen, bzw. vorliegen können.
Wenn Sie sich Ihren Commodore-Computer einmal etwas näher betrachten, können Sie feststellen, daß er einerseits für Sie zwar ein unheimlich intelligenter Kasten ist, andererseits aber eigentlich fast nichts kann. Denn wenn Sie ihn einschalten, dann meldet er sich auf dem Bildschirm zwar mit seiner Einschaltmeldung, aber ansonsten scheint er überhaupt nichts mehr zu tun. Falls Sie Ihren Computer eben erst ausgepackt haben, dann sollten Sie erst einmal das Handbuch studieren, um zu dem Punkt zu kommen, an dem auch für Sie die Einschaltmeldung zu sehen ist, denn das Handbuch wollen wir hier eigentlich nicht nochmal abdrucken, das haben Sie ja schon. Also nochmal, außer der Einschaltmeldung scheint Ihr Computer nichts weiter tun zu wollen. Die Wirklichkeit sieht aber ganz anders aus. Ihr Computer ist trotz dieses äußeren Anscheins in seinem Inneren sehr rege, wie Sie später noch selbst feststellen werden.
In diesem – anscheinend ruhenden – Zustand baut er nämlich das Bild, das Sie sehen, auf und außerdem wartet er auch dauernd darauf, daß Sie ihm etwas Sinnvolles zu arbeiten geben. In erster Linie kann dies erst einmal über die Tastatur geschehen. Und da gibt es bereits zwei Möglichkeiten. Erstens, Sie geben einen Befehl ein oder zweitens, Sie schreiben ein Programm.

DIREKT- UND PROGRAMMMODUS

Der erste Fall wird als sogenannter Direktmodus bezeichnet. All das, was Sie eintippen und dann mit der RETURN-Taste abschließen, versucht er sofort auszuführen. Vorausgesetzt, Sie geben keine Zahlen zu Beginn ein. Allerdings ist jeder Computer sehr heikel, er versteht nämlich nur das, was er will (besser ausgedrückt, was er kann). Das ist der Grund, daß es des öfteren (nach sinnlosen Eingaben) das manchmal schon richtig terrorisierende „Syntax Error" (es wurde von uns bereits schon in Syntax-Terror umgetauft) auf dem Bildschirm zu sehen gibt.
Es ist also erforderlich, sich im Dialog mit dem Computer auf das zu beschränken, was dieser auch erkennt. Die Befehlsform (= Syntax) muß also stimmen, denn sonst weist er Sie mit der bereits angeführten Fehlermeldung immer wieder darauf hin, daß er Sie nicht richtig versteht. Und da ist er wirklich sehr hartnäckig bzw. sogar richtiggehend stur. Eine solche Fehlermeldung erscheint aber nur dann direkt, wenn Sie im schon bereits erwähnten Direktmodus arbeiten. Der Computer erkennt aber automatisch, ob es sich um den Direktmodus handelt. Sind Sie nämlich im Programmiermodus, dann kommt vor dem ersten Befehl erst einmal eine Zeilennummer und daran erkennt er, daß er nicht sofort reagieren muß, sondern diese Eingaben nun als Programm zu verstehen hat. Er führt also Befehle, die am Anfang mit einer Zeilennummer versehen sind nicht aus, sondern legt diese als Programm ab.
Wenn Sie also eingeben:

PRINT 5+7 (RETURN-Taste)

antwortet er sofort mit 12. Sie haben ihn also – in einer für Rechner üblichen Form – aufgefordert, die Summe aus den beiden Zahlen zu berechnen und sofort auszugeben. Schreiben Sie aber:

100 PRINT 5+7 (RETURN-Taste)

dann gibt er Ihnen kein Ergebnis aus, denn er hat 

LERNEN SIE IHREN COMPUTER KENNEN

erkannt, daß Sie nun eine Programmzeile geschrieben haben. Das Ergebnis dieser kleinen Rechenaufgabe erscheint also erst, wenn Sie RUN eingeben und dann – wie immer zum Abschluß von Eingaben – die Returntaste drücken.
Daran, daß der Computer bemerkt was Sie wollen, nämlich ob Sie programmieren oder ihn direkt zu etwas auffordern wollen, sehen Sie, daß er eine gewisse Art von Grundintelligenz besitzt. Allerdings keine Art, die der menschlichen entspricht, sondern eben seine eigene Maschinenintelligenz. Was ihm dazu verhilft, das erfahren Sie später noch ganz ausführlich.

SEHEN WIR UNS DEN COMPUTER NOCH ETWAS GENAUER AN

So ein Computersystem besteht aus verschiedenen Teilen. Zunächst einmal hat er ein Gehäuse, in welchem ein großer Teil des gesamten Systemes untergebracht ist. So befinden sich einige Anschlüsse, der Netzschalter usw. an Ihrem Gerät.
Je nachdem, welchen Typ und welche Ausbaustufe Sie besitzen, kommt nun noch ein separates Netzteil, eine Diskettenstation, ein Joystick usw., hinzu.
Das, was Sie als Computer gekauft haben, ist selbst in der Minimalkonfiguration ein komplett lauffähiges Rechnersystem, das Sie dann auch noch nach Wunsch erweitern können. Die Commodore-Computer haben zwar eine Tastatur integriert, aber Sie müssen selbstverständlich auch noch ein Sichtgerät anschließen. Dies kann ein Fernsehgerät sein oder auch ein echter Monitor, der natürlich auch ein besseres Bild liefert. Aber soweit sind Sie ja bereits, das zu wissen, denn Sie haben Ihren Computer ja bestimmt schon ausgepackt und Ihre ersten Gehversuche unternommen.
Damit Sie auch verstehen. wo die groben Unterschiede der drei Computer liegen, die wir in dieser Serie behandeln, wollen wir ganz kurz darauf eingehen.
Der eigentliche Unterschied zwischen dem C16 und C116 liegt nur in der Tastatur und im Gehäuse. Die komplette Elektronik ist trotz des vorhandenen Größenunterschiedes gleich. Die Computerplatine hat zwar eine andere Größe und Bauform, aber elektronisch stimmen sie überein. Der C16 besitzt eine relativ gute Schreibmaschinentastatur, während der C116 kaum einzusetzen und auch die Programmierung (das Schreiben von Programmen) nicht so komfortabel ist wie beim C16. Diese beiden Computer wiederum sind ansonsten kompatibel zum Plus4, der gegenüber den beiden anderen auch noch mehr RAM-Speicher besitzt. Ein weiterer Unterschied ist auch noch der, daß der Plus4 einen 8-Bit-Userport besitzt und auch noch bereits eingebaute Software vorhanden ist. Ansonsten aber sind die drei Typen derart übereinstimmend, daß alle Software unter der Voraussetzung, daß genügend RAM-Speicher zur Verfügung steht, auch auf allen drei Typen läuft.
Auf weitere kleine Detailunterschiede wollen wir im Moment nicht eingehen.
So, evtl. war das schon zuviel des Guten und deshalb nun erst einmal grundsätzliche Erklärungen allgemein und speziell natürlich zu Computern. Wir können unmöglich immer dann, wenn ein Ihnen vielleicht nicht geläufiger Begriff auftaucht, sofort die Erklärung geben, denn sonst würden wir ja ein Lexikon schreiben und das ist nun mal nicht der Sinn dieser Artikel. Falls also Begriffe nicht sofort erklärt werden, dann sind diese im Augenblick auch nicht so wichtig, wie z.B. der Begriff „Userport".
Die erforderliche Erklärung hierzu geben wir also dann, wenn er irgendwann einmal eingesetzt wird.

DER COMPUTER, DAS UNBEKANNTE GERÄT

Zunächst einmal sollten Sie wissen, daß ein Computer nicht ein „kleiner grauer Kasten" ist, sondern daß er ein umfangreiches und durchaus sehr reges Innenleben besitzt. Das Herz eines Computers ist zunächst einmal der Prozessor. Dieser wird oft auch mit CPU (Central Prozessing Unit = Zentraleinheit) oder Mikroprozessor bezeichnet. Wie dieser Prozessor intern aufgebaut ist, braucht uns noch nicht weiter zu interessieren. Was Sie aber wissen sollten ist, daß sich ein solcher Prozessor mit den an ihm angeschlossenen weiteren elektronischen Bauteilen unterhalten (kommunizieren) muß. Denn es bringt überhaupt nichts, wenn jemand „denkt" und arbeitet – so wie der Prozessor – aber die Ergebnisse dieses Prozesses nicht in irgendeiner Weise verwertet werden.

RAM – BIT – BYTE

Sie sollten sich im Augenblick also vor allem merken, daß der Prozessor im Computer viel Arbeit verrichtet und daß alle Programme vom Prozessor abgearbeitet werden. Ihm zur Seite stehen aber noch andere Elektronikbauteile, die ihm tatkräftig behilflich sind.
Wie Sie vielleicht schon dem Handbuch oder dem Werbeprospekt entnommen haben, besitzt ein Computer soundsoviel RAM-Speicherplatz. Nehmen wir als Beispiel den C16. Dieser besitzt in seiner Grundausstattung 16 KByte-RAM.
Was heißt das? Zunächst einmal das Wort Speicher. Es weist in seinem wörtlichen Sinne schon fast darauf hin, was es ist. Ein Speicher wird üblicherweise dazu benutzt, irgendwelche Dinge zu verwahren, also aufzuheben. Ein elektronischer Speicher ist nicht anders zu betrachten. Auch in ihm wird aufgehoben, also gespeichert. Oft spricht man auch davon, daß eine Speicherzelle geladen wird, wenn man dort einen Wert ablegt.
Wenn Sie sich einen riesigen Schrank vorstellen, der viele kleine Schubladen hat (im Falle des C16 = 16384), dann haben Sie eine etwas bessere Vorstellung, was Speicherbereich oder -stelle bedeutet. Der gesamte Speicherbereich wäre also der Schrank und eine Speicherstelle entspräche einer Schublade. Um beim Schrankvergleich zu bleiben: In jede Schublade können Sie nur eine ganz bestimmte Menge eines Gutes legen, denn wenn die Schublade voll ist, dann ist sie eben voll. Genauso ist es auch bei einem elektronischen Speicherplatz. Wenn dieser voll ist, dann geht nichts mehr hinein. Ein wenig klemmt dieser Vergleich, denn bei einer Schublade können Sie, solange sie geöffnet ist, auch noch aufhäufen und die Schublade überladen. Wenn Sie die Schublade aber in den Schrank einschieben, dann sehen Sie, wie voll eine Schublade normalerweise sein kann!
Sie wissen nun also, der RAM-Speicher besteht aus RAM-Speicherstellen, in denen etwas stehen

kann und die nur eine bestimmte Menge aufnehmen können.
RAM ist die Abkürzung der englischen Bezeichnung Random Access Memory. Frei übersetzt: Speicher für den wahlfreien Zugriff. So wie Sie in eine Schublade etwas hineinlegen können, ist es auch möglich, wieder etwas herauszunehmen. Das gilt auch für RAM-Speicherstellen. Auch in diese kann etwas geschrieben, aber auch wieder ausgelesen werden. Und genau das ist das Kennzeichen eines RAM-Speichers. Im Gegensatz hierzu gibt es auch ROM-Speicher. Wie bei den meisten Bezeichnungen stammt auch diese Abkürzung aus dem englischen und stellt die Anfangsbuchstaben von Read Only Memory dar.

BYTE = ELEKTRONISCHE SCHUBLADE

Die Englischkundigen unter Ihnen werden nun bereits die Lösung erahnen: Dieser Speicher läßt sich nicht beschreiben, sondern nur auslesen. Wie aber kann etwas ausgelesen werden, wenn nicht geschrieben werden kann? Das hat der Hersteller bereits für Sie übernommen, dieser hat bereits das, was gelesen werden soll, eingeschrieben. Der Inhalt ist nicht mehr zu verändern, es sei denn, Sie zerstören das ROM, aber damit dann auch den sinnvollen Inhalt eines derartigen Speichermediums. Eine Zwitterstellung zwischen RAM und ROM nimmt das EPROM ein. Auch hier wieder die Ableitung aus der englischen Bezeichnung: Erasable Programmable Read Only Memory. Die Übersetzung lautet: Löschbarer programmierbarer Nurlese-Speicher. Es handelt sich also um einen Speicher, der normalerweise nur ausgelesen werden kann. Durch intensive Bestrahlung mit UV-Licht lassen sich die Inhalte wieder löschen und mittels spezieller Programmiergeräte können neue Daten wieder eingeschrieben werden.
Nun wissen Sie also, was verschiedene Speichertypen sind. Doch was paßt denn eigentlich in eine solche elektronische Schublade? Das ist schnell beantwortet: Werte zwischen Null und 255. Das sind also genau 256 verschiedene Zahlen. Ein fürchterlich „krummer“ Wert, finden Sie nicht?

```
MONITOR
   PC   SR AC XR YR SP
; FFFF 00 FF FF FF F8
M
>0000 0F C8 81 00 00 00 00 22 :.H......"
>0008 22 1C 00 FA 00 00 00 01 :"..◆....
>0010 00 98 00 00 64 00 19 16 :.....-..
>0018 00 02 FC FC 01 93 FC 00 :..▒▒..▒.
>0020 00 00 01 02 44 A4 00 00 :....D$..
>0028 00 00 00 01 10 1A 14 1A :........
>0030 14 1A 14 00 FD FE FC 00 :....|π▒.
>0038 FD 63 FF 01 02 9E 07 D2 :|-◥....R
>0040 07 00 10 0B 12 42 52 24 :.....BR$
>0048 14 04 04 02 02 00 00 00 :........
>0050 FF FF 00 00 4C BB 00 00 :◥◥..L;..
>0058 21 14 1A 13 00 00 00 00 :!.......
```

Bild 1: Mit dem Monitor ist zu sehen, daß in den unteren Speicherstellen etwas steht.

Aber auch das hat seinen Grund. Wir nehmen wieder unsere Schublade und unterteilen diese durch Trennwände und zwar so, daß insgesamt acht Fächer entstehen. Des weiteren gehen wir nun davon aus, daß in jedes Fach genau ein Teil gelegt werden kann. Dadurch ergeben sich insgesamt 256 Kombinationsmöglichkeiten zwischen den Zuständen Schublade leer und Schublade voll. Für die mathematisch interessierten Leser: Die Kombinationsmöglichkeit ist 2 hoch 8. Was wir bisher nun für die Schublade behandelt haben, ist auch bei einem Byte so. Das war nun wieder überaus gemein von uns, denn nun , wir haben Byte noch gar nicht geklärt, aber hier kommt die Erklärung: Ein Byte können Sie der Einfachheit halber mit einer Schublade vergleichen. Die Untereinheit in der Schublade ist ein Bit und dieses entspricht einem Fach in der Schublade. Ein Byte hat also acht Bit!
So, nun haben wir das alles rein oberflächlich betrachtet. Aber recht viel tiefer brauchen wir an dieser Stelle eigentlich gar nicht zu gehen. Die kleinste Einheit, die bei einem Computer berücksichtigt werden kann, ist ein Bit. Um nun möglichst anschaulich zu werden, vergleichen wir ein Bit einmal mit einem ganz normalen Schalter. Ein Schalter ist nichts anderes wie eine Schranke bei der Eisenbahn. Nur gibt es bei dieser außer den Stellungen geschlossen oder offen auch noch Zwischenstellungen. Diese sind aber nur deshalb vorhanden, weil eben nichts in der Zeit Null von einem Zustand in den anderen gehen kann. Im Falle der Schranke dürfte das auch gar nicht sein, denn was wäre denn, wenn beim Schließvorgang gerade ein Fahrzeug unter der Schranke steht? Scherz beiseite, eine gewisse Zeit, um den Zustand zu wechseln, wird immer benötigt, und sei sie noch so kurz. Bei elektronischen Schaltern ist diese Zeit so kurz, daß man einfach idealisiert und deswegen sagt, es gibt nur zwei Schalterstellungen: „ein“ und „aus“. Bei normalen Lampen für die Wohnzimmerbeleuchtung ist dies genauso. Entweder ist das Licht ein oder aus. Lichtdimmer sind ausgenommen, denn das sind keine Schalter, sondern Stellglieder! Die Zeit, bis sich der Lampenglühfaden bis zur Weißglut aufheizt, ist dabei so kurz, daß wir überhaupt nicht bemerken, daß es auch Übergangszustände gibt. Ganz wichtig ist also die Denkweise, daß ein Schalter nur offen oder geschlossen sein kann. Ein dreiviertelgeschlossen oder halboffen gibt es bei einem Schalter nicht.
Ein Bit ist – wie schon angeführt – ein „Schalter“. Dieses Bit kann also nur im Zustand ein oder aus sein. Man spricht auch davon, ein Bit sei gesetzt (= ein) oder nicht (= aus). Nochmals zur Erinnerung, ein Byte hat acht Bit. Nun sind wir hoffentlich soweit, daß die Bezeichnung Bit oder Byte keine Fremdworte mehr für Sie darstellen.

KILO IST NICHT IMMER TAUSEND!

So wie es für kleinere Mengen Untereinheiten gibt, so gibt es für größere Mengen auch „Übereinheiten“. Sicherlich ist Ihnen bekannt, daß es bei unserem Längenmaß „Meter“ auch die Untereinheit Dezimeter, Zentimeter, Millimeter usw. gibt. Sollen große Werte angegeben werden, so benützen wir beispielsweise die Kilometer. Das „Kilo“ vor der Maßeinheit bedeutet in diesem Falle eine Multiplikation mit 1000. Darüber hinaus gibt es dann noch weitere Vielfache, die beispielsweise dann durch den Vorsatz Mega, Giga usw. gekennzeichnet werden. Bei den Computern ist das ebenso, auch hier gibt es die Vielfachen, ➜

nämlich Kilo, Mega usw. Während aber bei den sonst üblichen „Kilos" diese Kennzeichnung durch Kleinschrift (kg, km, kp etc.) des ersten Buchstabens erfolgt, geschieht dies bei der Computerei durch Großschrift. Richtig muß es also heißen KByte und nicht kByte! Der Grund ist ganz einfach: Das „Computerkilo" sagt nämlich nicht aus, daß mit 1000 multipliziert (malgenommen) werden muß, sondern mit 1024!
Dies ergibt sich aus der Computerei selbst. Wenn Sie sich bitte erinnern wollen, mit acht Bit können 256 Zustände signalisiert werden. Wenn Sie die 256 mit 4 multiplizieren, dann erhalten Sie den Wert 1024. Diese ganze Angelegenheit rührt also daher, daß eben die maximale Kombinationsmöglichkeit bei einem Acht-Bit-Prozessor immer wieder auf die Basis Acht zurückführt. Frühere Acht-Bit-Prozessoren konnten auch nur einen Adressbereich von acht Bit ansprechen und dieser Bereich wurde „page" (engl.: Seite) genannt. Durch verschiedene Techniken war man später aber auch in der Lage, mehrere Seiten anzusprechen. Dadurch ergab sich dann, daß man vier Seiten eben ein Kilo nannte. Nun wissen Sie auch, woher diese krummen Werte stammen, und Sie können nun gleich einmal nachrechnen, wieviel RAM-Speicherbytes Ihr Computer besitzt.
Der C16/116 besitzt 16 KByte RAM, also 16 mal 1024 = 16384 und der Plus4 besitzt 64 KByte RAM, also 64 mal 1024 = 65536.
Soviel Speicherplatz kann bei Ihrem Computer beschrieben und gelesen werden. Allerdings steht Ihnen diese Speicherkapazität nicht voll zur Verfügung, denn der Computer selbst braucht davon auch einen ganz gehörigen Anteil. Um das zu verstehen, müssen wir uns nun den Computeraufbau etwas näher ansehen.
Als Beispiel nehmen wir den C16/116.
Wenn wir nun in die untersten Schubladen hineinsehen, dann stellen wir fest, daß in diesen bereits etwas steht, obwohl wir nichts hineingeschrieben haben!

DER ERSTE EINSATZ VON TEDMON

Wie, Sie können nicht in diese Speicherstellen sehen? Doch, es ist ganz einfach, denn Ihr Computer stellt Ihnen nämlich bereits das Werkzeug hierfür zur Verfügung. Die „Lesebrille", die wir im Augenblick brauchen, können Sie ganz einfach durch die Eingabe von Monitor aufrufen. Das heißt, Sie brauchen Ihren Computer nur einzuschalten und dann über die Tastatur einzugeben:

MONITOR (RETURN)

Das eingeklammerte RETURN bedeutet dabei, daß Sie nicht das Wort – Buchstabe für Buchstabe – eintippen sollen, sondern ist nur der Hinweis, daß Eingaben mit der RETURN-Taste abgeschlossen werden müssen.
Nachdem Sie diesen Befehl eingegeben haben, meldet sich TEDMON – so heißt dieser Monitor – folgendermaßen:

```
MONITOR
     PC   SR  AC  XR
;  XXXX XX  XX  XX
   YR  SP
   XX  XX
```

außerdem ist der Cursor zu sehen, und das bedeutet, daß der Monitor auf eine Eingabe von Ihnen wartet. Die X-Zeichen sind bei Ihnen allerdings Zahlen, auf die wir im Augenblick noch nicht näher eingehen wollen. Das folgt zu einem späteren Zeitpunkt, wenn Sie schon etwas vertrauter mit Ihrem Computer sind. Wir wollen im Moment ja nur nachsehen, was in verschiedenen RAM-Speicherstellen Ihres Computers steht. Geben Sie deshalb den Buchstaben "M" ein und betätigen Sie die Return-Taste. Die Antwort des Computers ist dann ein zwölf Zeilen langer Speicherauszug als Hexdump und ASCII-Darstellung. (Siehe Bild 1) Bevor wir darauf näher eingehen, müssen erst einmal die Begriffe Hexdump und ASCII geklärt werden.

```
DER COMMODORE-ZEICHENSATZ FUER
DIE CHR$-CODE VON 32 BIS 127

 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFG
HIJKLMNOPQRSTUVWXYZ[£]↑←

   UND FUER 160 BIS 255

DAS LISTING:

10 PRINT"DER COMMODORE-ZEICHENSATZ FUER
20 PRINT"DIE CHR$-CODE VON 32 BIS";
30 PRINT" 127":PRINT:FORI=32 TO 127
40 PRINTCHR$(I);:NEXT:PRINT:PRINT
50 PRINT"    UND FUER 160 BIS 255":PRINT
60 FOR I=160 TO 255:PRINTCHR$(I);:NEXT
70 PRINT:PRINT:PRINT"DAS LISTING:":LIST
←
```

Bild 2: Die Hardcopy zeigt das Programm und seine Auswirkungen.

Dump ist die englische Bezeichnung von „auskippen" und damit ist die Ausgabe von Speicherstelleninhalten gemeint. Die Bezeichnung "Hex" rührt daher, daß die Ausgabe im sogenannten Hexadezimalcode geschieht.
Der Hexadezimalcode müßte korrekterweise eigentlich Sedezimalcode heißen, aber es hat sich zwischenzeitlich so eingebürgert, daß jeder nur von Hexadezimal- (oder kürzer: Hexcode) spricht. In der Schule haben wir vor allem Rechnen mit der Basis 10 erlernt und deshalb fällt es uns manchmal sehr schwer, mit anderen Zahlensystemen umzugehen.
Aber nach einiger Einarbeitung geht das dann fast genauso schnell wie mit der Basis 10. Übrigens arbeiten wir bei unseren älteren Maßeinheiten auch nicht immer mit der Basis Zehn.
Beispiele: Dutzend, Paar, Stunden/Tage. Damit Sie die Scheu vor anderen Zahlensystemen verlieren: Sie wissen doch, wieviele Einzelschuhe fünf Paar und bestimmt auch wieviel Stück ein halbes Dutzend Weinbergschnecken sind? Ganz bestimmt können Sie auch sagen, wieviele Minuten eineinhalb Stunden entsprechen. Sie sehen, alles reine Übungssache.
Wer anfangs trotzdem Probleme mit anderen Zahlensystemen hat, nicht verzweifeln, Sie schaffen das noch. Nun erst einmal zum Unterschied zwischen der dezimalen und hexadezimalen Schreibweise.

VER"HEX"TE ZAHLEN

Unser dezimales Zahlensystem ist folgendermaßen aufgebaut: Wenn wir bei Null beginnen, dann endet unsere einziffrige Darstellung bei Neun. Danach geht es dann zweistellig weiter und das wiederholt sich immer weiter so. Wenn man die Hexadezimalzahlen be-

trachtet, geht das in ähnlicher Weise vonstatten. Auch dabei beginnt das Zählen bei Null, die einziffrige Darstellung endet aber nicht bei Neun, sondern geht weiter über A, B,C,D bis F. Die Zeichen A bis F sind bei Hexadezimalzahlen also keine Buchstaben, sondern stellen die Zahlen von 10 bis 15 dar.

Eine kleine Tabelle gibt Ihnen dabei einen etwas besseren Überblick:

Dezimal	Hexadezimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
usw.	

Diese Zahlen sind für Sie bis zur Hexzahl von $FFFF von Interesse, denn &FFFF entspricht der Dezimalzahl 65536 und das ist der Speicherumfang, der von einem Acht-Bit-Prozessor ohne irgendwelche Tricks verwaltet werden kann.
Die Kennzeichnung, daß es sich um eine Hexzahl handelt, erfolgt bei der Computertechnik meist dadurch, daß entweder das kaufmännische Und-Zeichen (&) oder das Stringzeichen ($) der eigentlichen Zahl vorangestellt wird. Es gibt aber auch noch andere Darstellungsweisen. So wird auf eine derartige Zahl auch durch ein am Ende angehängtes “h“ hingewiesen. Diese Kennzeichnungen sind alle gleichbedeutend.
Nun verstehen Sie wahrscheinlich schon besser, was der Monitor auf dem Bildschirm darstellt, wenn er aufgerufen und die Funktion “M“ (Memory-Dump) ausgewählt wird. Er zeigt links immer einen Winkel und dann die Adresse. Daneben folgen dann acht Hexadezimalzahlen.
Diese Zahlen sind der Inhalt der dargestellten Speicherstellen. Rechts daneben sehen Sie dann noch die ASCII-Zeichen. Diese entsprechen ebenfalls den Inhalten. Die nächste Zeile ist wieder der ersten entsprechend, am Anfang die Speicheradresse usw. Wenn Sie im Monitor eingeben “M 1000“, dann sehen Sie ein evtl. vorhandenes Basic-Programm und können sogar einiges davon lesen. Aber nicht alles, denn da gibt es noch einige Besonderheiten, die wir später noch ansprechen werden. (Siehe Bild 3)

```
MONITOR
   PC   SR AC XR YR SP
; FFFF  00 FF FF FF F8
M 1000
>1000 00 26 10 0A 00 99 22 44 :.&...."D
>1008 45 52 20 43 4F 4D 4D 4F :ER COMMO
>1010 44 4F 52 45 2D 5A 45 49 :DORE-ZEI
>1018 43 48 45 4E 53 41 54 5A :CHENSATZ
>1020 20 46 55 45 52 00 47 10 : FUER.G.
>1028 14 00 99 22 44 49 45 20 :..."DIE
>1030 43 48 52 24 2D 43 4F 44 :CHR$-COD
>1038 45 20 56 4F 4E 20 33 32 :E VON 32
>1040 20 42 49 53 22 3B 00 61 : BIS";.♠
>1048 10 1E 00 99 22 20 31 32 :...." 12
>1050 37 22 3A 99 3A 81 49 B2 :7":.:.I2
>1058 33 32 20 A4 20 31 32 37 :32 $ 127
```

Bild 3: Ein Teil des Programmes zur Commodore-Zeichen-Ausgabe als Hexdump.

ASCII ist (wie üblich) eine Abkürzung von American Standard Code for Information Interchange und ist eine Normung für Codes, die bei Computern verwendet werden. Wie Sie nun bereits wissen, kann eine Speicherstelle als maximalen Wert die Zahl 255 (dezimal) aufnehmen. Aber die ASCII-Zeichen sind leider nur bis 127 genormt. (Commodore weicht aber von diesem Standard ab. Darauf gehen wir im nächsten Heft noch näher ein!) Zeichen, deren Codes größer als dieser Wert sind, sind sowohl von Computertyp zu Computertyp, als auch von Drucker zu Drucker unterschiedlich. Das führt dann zu Problemen, wenn „fremde“ Drucker an einen Computer angeschlossen werden sollen. In einigen Fällen treten aber auch Probleme auf, wenn Programme von einem Computertyp auf einen anderen übertragen werden sollen.
Die Zeichen bis chr$(127) sind also meist auf allen Geräten identisch, aber darüber hinaus dann nicht mehr. Nun haben Sie bereits gesehen, wie die Ausgabe verschiedener ASCII-Zeichen auf Ihrem Computer erreicht werden kann.
Sie brauchen nur einzugeben:

PRINT CHR$(x)

Als x können Sie Zahlen bis 255 nehmen, aber Sie sollten es nicht tun, solange Sie noch nicht weitergelesen haben, denn manche Zeichen lassen sich nicht darstellen, sondern sind Steuerzeichen, die ganz bestimmte Auswirkungen haben. So löscht beispielsweise der Befehl PRINT chr$(145) den Bildschirm, chr$(19) setzt den Cursor in die linke obere Ecke des Bildschirmes usw. Sie sehen, auch bei niedrigen Werten sind Steuerzeichen zu finden und deshalb sollten Sie auch keine Zeichen unterhalb von 32 verwenden. Diese dienen nur zur Steuerung und nicht zur Darstellung.
Damit Sie aber den Standard-Commodore-Zeichensatz auf dem Bildschirm einmal sehen, können Sie folgendes eingeben:

FORI = 32 to 127:PRINT chr$(i);:NEXTI
und
FORI = 160 to 255:
PRINT chr$(i);:NEXTI

Sie können aber auch ein kleines Programm eingeben. Bild 2 zeigt sowohl das Programm, als auch seine Auswirkungen.
Vorher müssen Sie allerdings den Monitor verlassen.
Beim nächsten Mal geht es dann weiter mit der Untersuchung der Programmablage und bereits einigen Tips und Kniffen.

DIE PROGRAMM-ABLAGE

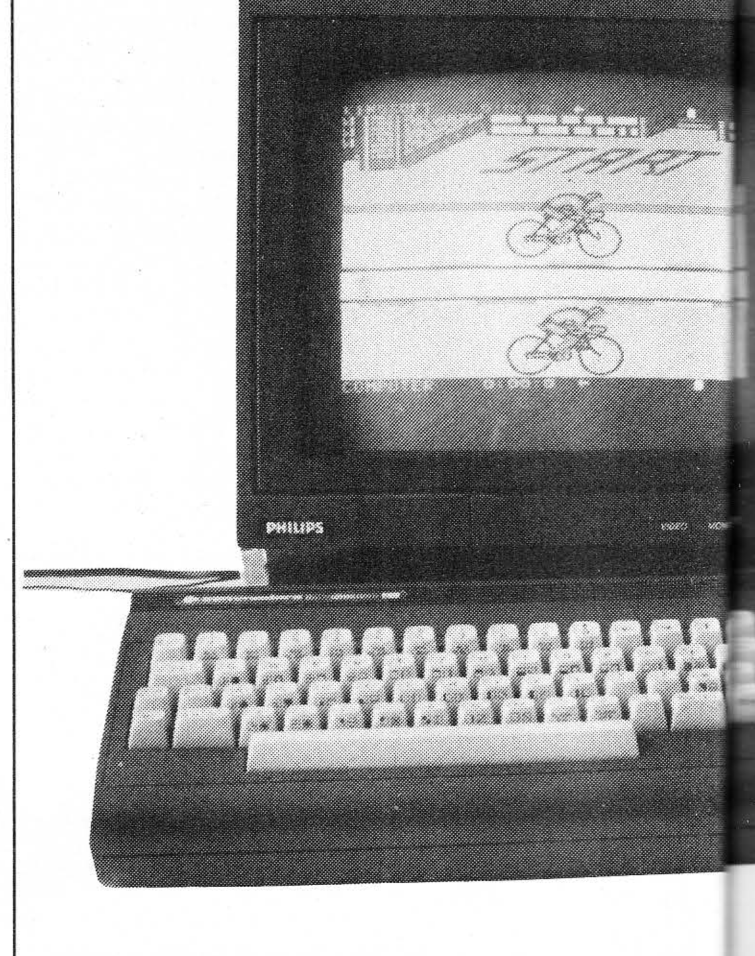
Bevor Sie sich dies nun genauer ansehen können, sollten Sie Ihren Computer zurücksetzen (den kleinen Reset-Taster am Computer kurz eindrücken). Schreiben Sie dann bitte folgendes kleines Programm:

```
10  REM TEST
20  PRINT "COMPUTER"
```

So, nun sehen Sie sich das, was Sie eben programmiert haben, einmal an. Dazu gleich wieder den TEDMON (so heißt der eingebaute Monitor) aufrufen.
Also:
MONITOR (RETURN)
und dann gleich
M 1000 (RETURN)
eingeben, schon sehen Sie Ihr kleines Programm. Wenn Sie die ASCII-Darstellung betrachten, können Sie sogar einige Textteile erkennen. Aber es geht ja noch viel weiter, Sie erkennen die Token, die Linkadressen usw. Näheres darüber erfahren Sie auf den nächsten Seiten.

LERNEN SIE IHREN COMPUTER KENNEN

Andeutungsweise hatten wir schon aufgezeigt, daß der Computer selbst — auch bei den Typen mit geringem RAM-Speicher viele der für größere Programme doch so dringend benötigten — Speicherplätze belegt. Aber das geht eben nicht anders, denn er muß sich ja merken können, welche Farben Sie wollen, wo sich der Cursor befindet, wo der Basicstart ist, wo er Variablen abgelegt hat und auch, wie er sie wieder finden kann, wenn Sie ihn danach fragen. Auch das, was Sie auf dem Bildschirm sehen, muß irgendwo im RAM gespeichert sein. Es gibt noch viele weitere Dinge, die er sich merken muß, und deshalb ist ein bestimmter Teil für Programme einfach nicht verfügbar. Wo er sich was merkt, darauf gehen wir später noch ein, sehen wir uns nun aber einfach einmal an, was der Commodore mit einem Programm, also mit den Eingaben, die mit einer Zeilennummer versehen wurden, anstellt.



Starkes Basic: Commodores „Kleine“

Wir hatten Ihnen bereits mitgeteilt, daß der Commodore im unteren Speicherbereich einige Informationen abgelegt hat, die er benötigt. Auch der Hinweis, wo Sie ein Basicprogramm zu suchen haben, stand bereits in der Commodore-Welt (Sonderheft 2/87). Zur Erinnerung nochmals der Hinweis: Ein Basicprogramm wird von ihm ab der Adresse S1000 (= dezimal 4096) abgelegt.

DIE PROGRAMMABLAGE

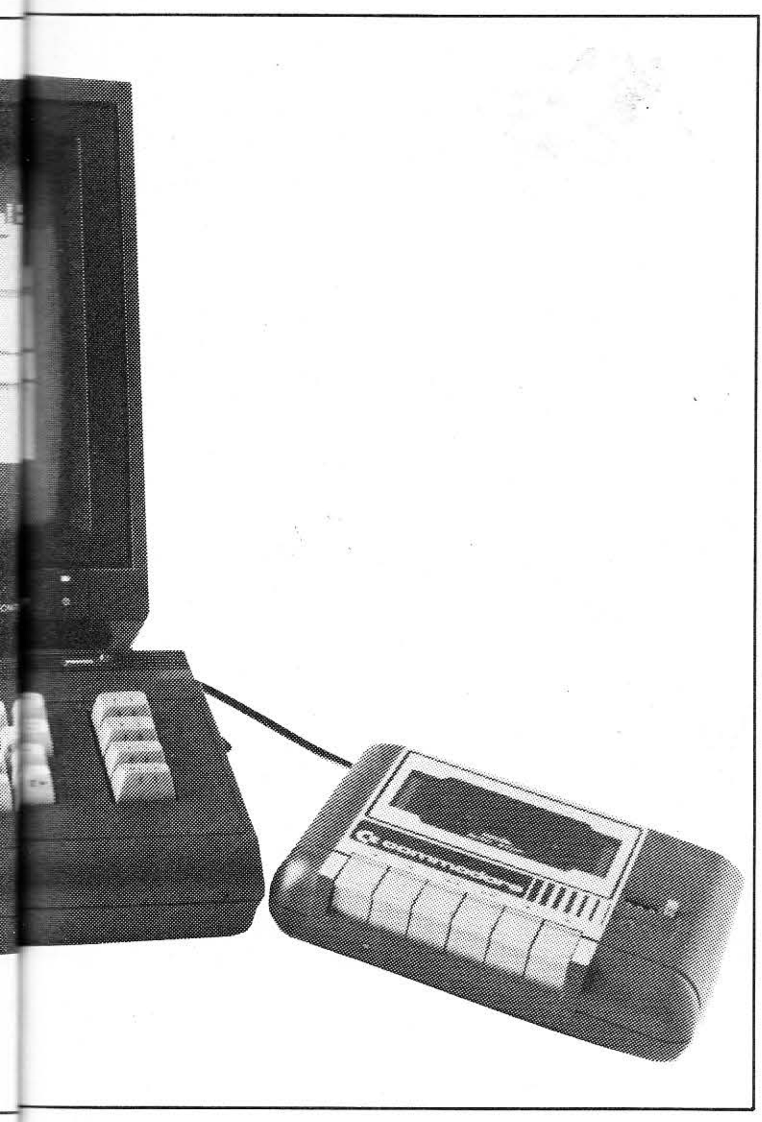
Haben Sie sich die Programmablage der beiden — im letzten Heft abgedruckten — Programm-Zeilen näher angesehen? Führten Ihre Bemühungen insofern auch zum Erfolg, daß Ihnen alle Einzelheiten klar wurden? Falls Sie diese zweite Frage mit ja beantworten können, dann sind Sie ja ein richtiger Computerprofi. Falls nicht, kein Problem, wir helfen Ihnen.
Schreiben Sie bitte folgendes kleines Programm:

```
10 REM TEST
20 PRINT"COMPUTER"
30 PRINT "PROGRAMM-
   ENDE"
```

So, nun sehen Sie sich das, was Sie eben programmiert haben, an. Dazu den TEDMON, also den eingebauten Maschinensprache-Monitor, aufrufen:
MONITOR (RETURN)
und dann gleich
M 1000 (RETURN)
eingeben, schon sehen Sie Ihr kleines Programm.
Bei der Betrachtung der ASCII-Darstellung können Sie einige Textteile erkennen. Da diese Teile aber ja Klartext sind, brauchen wir uns nicht weiter damit zu beschäftigen.
An dieser Stelle gleich ein kleiner Hinweis, wie Sie sich Programmier-Arbeit erleichtern können. Sie können sich nämlich viele Tastendrucke sparen, wenn Sie wissen, daß die Commodore-Computer die hervorragende Eigenschaft haben, auch „Kurzschrift“ zu verstehen. Fast alle Befehlsworte

COMMODORE-STENO

können abgekürzt eingegeben werden. Diese Kurzform besteht aus dem ersten Buchstaben des Befehlswortes in ganz normaler Schreibweise und der geshifteten (halten Sie die SHIFT-Taste fest) Eingabe des zweiten Buchstabens. Bei manchen Befehlen ist es erforderlich, daß drei Buchstaben eingegeben werden müssen und der dritte dann „geshiftet“ wird. Sie können das ruhig einmal ausprobieren. Bei Eingaben können so viele Tastendrucke „gespart“ werden. Zum Monitoraufruf können Sie also auch M und (Shift) O eingeben und dann die Return-Taste drücken.
Nun zu den Untersuchungen. Nachdem der Monitor aufgerufen wurde, bitte M 1000 eingeben. Nun sehen Sie das zuvor eingegebene Programm, bzw. das, was der Computer daraus gemacht hat (Bild 1).
In der ersten Speicherstelle ($1000) befindet sich prinzipiell eine Null. Die nächsten beiden Hexadezimalzahlen tauschen Sie einfach aus und bilden daraus dann eine vierstellige Hexzahl. Damit haben Sie die Linkadresse zur nächsten Basiczeile



erhalten. Die Linkadresse wird auch Verweiszeiger genannt, da sie auf den nächsten Basiczeilenbeginn hinweist. Die Abspeicherung geschieht dabei immer in sogenannter Low-Byte/High-Byte-Konfiguration.
Wenn Sie sich die zweite und dritte Hexzahl in der Zeile ansehen, in der links >1000 steht, dann lesen Sie 0C 10. Die Hexzahl oC (dezimal 12) ist das Low-Byte, also das niederwertige Byte und 10 (dezimal 16) das höherwertige. Da bei Zahlendarstellungen aber die „höheren Stellen" links zu stehen haben, ergibt sich daraus die Hexzahl $100C.
Dort beginnt also die nächste Basiczeile. Sehen wir uns aber erst einmal die erste Zeile weiter an.

Nach der Verweisadresse (manchmal auch Koppeladresse genannt) machen Sie das gleiche nun wieder. Setzen Sie wiederum die Hexzahl umgekehrt zusammen, dann erhalten Sie $000A. Da führende (also links stehende Nullen) weggelassen werden können, erhalten Sie den hexadezimalen Wert"A".
Wenn Sie nun die kleine Übersicht aus dem vorhergehenden Artikel zu Rate ziehen, dann können Sie bereits sehen, daß dies der Dezimalzahl 10 entspricht. Vermutlich ahnen Sie nun schon, was diese Zahl zu bedeuten hat. Richtig, diese Zahl ist die Zeilennummer der ersten Basiczeile.
Die nächste Hexzahl müssen Sie nun für sich alleine betrachten. Der Wert ist $8F. Können Sie damit etwas anfangen? Diese Zahl ist der Basic-Befehl REM. Wenn Sie sich nun erstaunt fragen, warum keine 3 Bytes für das Wort benötigt werden, dann wird Ihnen die Lösung gleich hier präsentiert: Der Commodore schreibt Steno. Damit nicht noch mehr des meist ohnehin zu knappen Speicherplatzes belegt wird, und auch noch aus anderen Gründen, werden alle Befehlsworte codiert abgelegt und benötigen deshalb nur ein Byte im Speicher des Computers. Der Wert $8F ist der Code für REM. Diese Codes werden "Token" genannt. Ein Token ist also die Kurzform für ein Befehlswort.

ASCII UND HEXADEZIMAL

Die nächste Zahl ($20, dez.: 32) ist der Code der Leertaste (Space). Wenn Sie im Handbuch einmal nachsehen, finden Sie bei den Zeichencodes diese Werte. Nun können Sie gleich weitermachen. Das im rProgramm folgende Wort lautet TEST und dieses finden Sie ja auch in der ASCII-Darstellung.
Der ASCII-Wert (ASCII =American Standard Code for Information Interchange) für den Buchstaben "T" ist $54, für "E" ist er $45 usw.
Die ASCII-Werte können Sie sowohl im Handbuch nachlesen, als auch direkt am Computer erfragen.
Beispiel:
A$="T":PRINT ASC(A$)
(Return)
oder wer es gleich hexadezimal haben will:
A$="T":PRINT HEX$
(ASC(A$))

Bevor wir nun weitermachen, ein paar Worte zur Monitorausgabe.
Bei den eigentlichen Ausgaben, also ab der Zeile mit dem Wert 1000, sehen Sie links den kleinen Winkel. Man bezeichnet ihn als Prompt. Dann folgt die Speicheradresse mit dem Wert $1000. Wenn Sie wissen wollen, in welcher Speicherstelle beispielsweise 0A steht, brauchen Sie zum links stehenden Adresswert nur den Positionswert zu addieren, in dem die betreffende Zahl steht. Der erste Positionswert ist 0, der nächste 1 usw. bis zum Positionswert 7. Die letzte Adresse in der "Zeile" >1000 ist also $1007.
Folgerichtig ist deshalb die Adresse der nächsten Zeile $1008. Die letzte Adresse dieser Zeile wäre dann $100F usw.
Zur Verdeutlichung:

Adresse	Inhalt
1000	00
1001	0C
1002	10
1003	0A *
1004	00
1005	8F
1006	20
1007	54
1008	45
1009	53
100A	54

usw.
Der Wert 0A steht also in der Speicherstelle S1003. Diejenigen, die bisher noch nicht verstanden haben, daß es besser ist, sich die hexadezimale Denk- und Arbeitsweise anzugewöhnen, werden nun langsam wohl verstehen, daß dies günstiger ist, wenn man seinen Computer kennenlernen will.
So, nun machen wir wieder weiter. Bis zum letzten Buchstaben des Wortes Test konnten Sie ja alles selbst verfolgen.

ZEILEN- UND PROGRAMMENDE

Nach dem Wort Test steht in der folgenden Speicherstelle der Wert Null. Dies ist das Kennzeichen für Basiczeilenende. Das nächste Zeichen steht in der RAM-Speicherstelle $100C. Erinnern Sie sich noch, wohin der Verweiszeiger deutete? Richtig, genau auf diese Speicherstelle. Und nun beginnt das ganze wieder von vorne. Die ersten beiden Bytes sind der Verweiszeiger auf die nächste Basiczeile, dann folgt die



Zeilen-Nummer usw., usw.
Nun sind Sie bestimmt in der Lage, die zweite und dritte Programmzeile selbst zu analysieren. Eines sollten Sie aber noch wissen: Die Linkadresse der letzten Basiczeile deutet auf eine Null. Ihr folgt eine weitere Null. Das ist das Kennzeichen für den Computer, daß hier das Programm zu Ende ist. Da ein Basiczeilenende aber auch mit einer Null gekennzeichnet wird, befinden sich am Programmende immer drei Nullen hintereinander.
Die erste Hürde wäre nun geschafft. Nun dürfen Sie sich einmal richtig austoben. Mit dem TEDMON kann man ja nicht nur Speicherstellen ansehen, sondern auch verändern. Dies geht ganz einfach. Bewegen Sie mit den Cursor-Steuertasten den Cursor einfach auf die Speicherstelle, die Sie verändern wollen und geben dann das gewünschte Zeichen über die Tastatur ein. Wenn Sie in der Zeile fertig sind, drücken Sie Return und die von Ihnen eingegebenen Werte werden übernommen. Voraussetzung ist aber, daß diese gültig sind, also Hexziffern. Der Buchstabe "K" wird also bestimmt

EINE BASIC-ZEILE VERSCHWINDET

nicht akzeptiert. Je nach Eingabe ändert sich dann auch die ASCII-Darstellung. Falls Sie sich wundern sollten, weshalb es in der ASCII-Darstellung soviele Punkte gibt, obwohl die zugehörigen Hexzahlen unterschiedlich sind, dann erinnern Sie sich bitte an die Geschichte mit den Steuerzeichen! Würde der Commodore nämlich versuchen, diese auszugeben, dann würde nicht das Zeichen folgen, sondern der Befehl ausgeführt. Im Falle der Hexzahl $93 würde also der Bildschirm gelöscht werden!

```
10 REM TEST
20 PRINT"COMPUTER"
30 PRINT"PROGRAMM-ENDE"
READY.
M

MONITOR
   PC  SR AC XR YR SP
; FFFF 00 FF FF FF F8
M 1000
>1000 00 0C 10 0A 00 8F 20 54 :...... T
>1008 45 53 54 00 1C 10 14 00 :EST.....
>1010 99 22 43 4F 4D 50 55 54 :."COMPUT
>1018 45 52 22 00 31 10 1E 00 :ER".1...
>1020 99 22 50 52 4F 47 52 41 :."PROGRA
>1028 4D 4D 2D 45 4E 44 45 22 :MM-ENDE"
>1030 00 00 00 00 00 00 00 00 :........
```

BILD 1:HARDCOPY EINER PROGRAMM-ABLAGE

```
   PC  SR AC XR YR SP
; FFFF 00 FF FF FF F8
M 1000 1030
>1000 00 1C 10 0A 00 8F 20 54 :...... T
>1008 45 53 54 00 1C 10 14 00 :EST.....
>1010 99 22 43 4F 4D 50 55 54 :."COMPUT
>1018 45 52 22 00 31 10 1E 00 :ER".1...
>1020 99 22 50 52 4F 47 52 41 :."PROGRA
>1028 4D 4D 2D 45 4E 44 45 22 :MM-ENDE"
>1030 00 00 00 4D 00 00 00 00 :...M....
X
READY.
LIST

10 REM TEST
30 PRINT"PROGRAMM-ENDE"
READY.
RUN
COMPUTER
PROGRAMM-ENDE
```

BILD 2: MANIPULATION AM VERWEISZEIGER

```
MONITOR

   PC  SR AC XR YR SP
; FFFF 00 FF FF FF F4
M 1000 1030
>1000 00 0C 10 0A 00 8F 20 54 :...... T
>1008 45 53 54 00 1C 10 14 00 :EST.....
>1010 99 22 43 4F 4D 50 55 54 :."COMPUT
>1018 45 52 22 00 01 10 1E 00 :ER".....
>1020 99 22 50 52 4F 47 52 41 :."PROGRA
>1028 4D 4D 2D 45 4E 44 45 22 :MM-ENDE"
>1030 00 00 00 00 00 00 00 00 :........
X
READY.
```

BILD 3: DURCH 'RUECKSTELLUNG' DER LETZTEN LINKADRESSE AUF DEN PROGRAMMBEGINN ENTSTEHT EIN UNENDLICHES LISTING

Sie können nun beispielsweise die Texte ändern, auch die Zeilennummern und die Linkadressen.
In Bild 2 ist zu sehen, daß wir die Verweisadresse der ersten Basiczeile, die ja eigentlich auf den Beginn der zweiten zeigen sollte, auf die dritte gestellt haben. Die Zeile mit der Nummer 20 wird bei LIST nicht mehr ausgegeben! Bei RUN wird sie aber abgearbeitet. Ein kleiner Trick also, eine Basiczeile zu verstecken. Allerdings sind Sie auf dem Holzweg, wenn Sie glauben, damit einen Programmschutz realisieren zu können, denn erstens kann man sich mit dem TEDMON alles ansehen und ändern und zweitens kann durch Neunumerierung der Basiczeilen (Befehl: RENUMBER) dies wieder rückgängig gemacht werden.

EIN UNENDLICHES LISTING

Diese bisherigen Manipulationen haben alle keinen tieferen Hintergrund, helfen aber die Arbeitsweise eines Computers besser zu durchschauen. Aus diesem Grunde zeigen wir Ihnen nun eine weitere lustige Veränderung an einem Programm. Betrachten Sie bitte Bild 3. Dort haben wir den Verweiszeiger der letzten Zeile wieder auf den Beginn der ersten Zeile gestellt. Der Befehl LIST bewirkt deshalb eine dauernde Programmausgabe. Beim Programmlauf aber verhält sich der Computer so, als wäre das Programm nicht verändert worden. RENUMBER stellt auch hier wieder alles richtig.
Nun also eine weitere interessante Manipulation: Eine Basiczeile löscht sich selbst! Bild 4 zeigt, was zu tun ist. Die Zeile 20 mit dem "Sternchen" bis zum rechten Bildschirmrand ergänzen, und dann die Hexziffern $2A (=Sternchen) durch

$14 ersetzen. Wenn Sie dann LIST oder RUN eingeben, dann wird die Basiczeile 20 zwar ausgegeben, löscht sich aber wieder von alleine, denn $14 ist der Code für ..., sehen Sie doch bitte selbst im Handbuch nach, denn sonst müßten wir hier ja dessen Inhalt nochmals abdrucken. Auch andere Steuercodes können Sie zu den verschiedenartigsten Effekten ausnutzen. So, alle diese Übungen waren vor allem dazu gedacht, daß Sie die Programmablage kennen – und vor allem, daß Sie mit dem Monitor umzugehen lernten. Wir wollen Ihnen ja noch viel sinnvollere Speicherinhaltsveränderungen mittels des TEDMON aufzeigen, aber vorher sollten Sie eben etwas vertrauter mit diesem Werkzeug werden. Bisher haben wir nur zwei der möglichen Befehle des Monitorprogrammes eingesetzt, nämlich den für die Speicheranzeige (M für Memory-Dump) und den zum Verlassen dieses Hilfsprogrammes (X für eXit). Daß der Monitor aber noch einige mehr hat, können Sie sehr leicht im Handbuch nachlesen. Nach und nach werden wir Sie auch noch in die anderen Befehle einweisen, aber um es klarzustellen, unser Ziel ist es nicht, Ihnen einen Lehrgang für den Umgang mit TEDMON zu liefern, sondern wir wollen Ihnen dabei helfen, Ihren Computer besser kennenzulernen und vor allem auch zeigen, wie Sie trickreich die Eigenarten Ihres Commodore in Programmen ausnutzen können. Einen kleinen Vorgeschmack auf sinnvolle Manipulationen wollen wir Ihnen diesmal mit einem kleinen Programm zeigen.
Bild 5 zeigt ein kleines Listing, welches bereits von dem Erlernten Gebrauch macht. Es dient zur Ausgabe der Token und zur Ausgabe deren Codezahl.

```
10 REM TEST
20 PRINT"COMPUTER***********************
30 PRINT"PROGRAMM-ENDE"

READY.
M

MONITOR

   PC   SR AC XR YR SP
; FFFF  00 FF FF FF F8
M 1000 1040
>1000 00 0C 10 0A 00 8F 20 54 :...... T
>1008 45 53 54 00 32 10 14 00 :EST.2...
>1010 99 22 43 4F 4D 50 55 54 :."COMPUT
>1018 45 52 14 14 14 14 14 14 :ER......
>1020 14 14 14 14 14 14 14 14 :........
>1028 14 14 14 14 14 14 14 14 :........
>1030 14 00 47 10 1E 00 99 22 :..G...."
>1038 50 52 4F 47 52 41 4D 4D :PROGRAMM
>1040 2D 45 4E 44 45 22 00 00 :-ENDE"..
```

BILD 4: EINE ZEILE VERSCHWINDET!

```
100 run 160
110 rem ***************
120 print chr$(147)
130 poke dec("1018"),a:print a;
140 list 110:end
160 a=127:printchr$(147):print"bitte immer wieder die taste f1 druecken
170 key 1,"a=a+1:goto 110"+chr$(13)
```

Bild 5: Token- und Codeausgabe

Die Zeile 100 bewirkt, daß das Programm in Zeile 160 startet. Die Zeile 110 ist eine kleine Hilfszeile, in welcher nachher ein Basicbefehl steht. Dieser Befehl wird durch POKEn des Wertes der Variablen A etwa in die Mitte der REM-Zeile 110 geschrieben.
Zeile 130 führt den POKE-Befehl aus und außerdem wird der Wert für A auf dem Bildschirm ausgegeben. Die Zeile 140 bewirkt, daß die Zeile 110 gelistet wird und beendet dann den Programmlauf.
Da bei einem Programmstart ein Neustart in Zeile 160 bewirkt wird, werden die Zeilen 110 bis 140 zunächst übergangen. In Zeile 160 wird der Variablen A der Wert 160 zugewiesen. Danach wird der Bildschirm gelöscht und außerdem wird mitgeteilt, daß die Funktionstaste F1 immer wieder betätigt werden soll.
Die letzte Zeile belegt die Funktionstaste mit einem Zeichenstring.
Dieser Zeichenstring lautet nach dem Programmstart:
A=A+1:GOTO 110
(CHR$13)
Wenn Sie nun also die Funktionstaste F1 drükken, wird dieser String auf dem Bildschirm ausgegeben. Das angehängte CHR$(13) ist der Befehl für Carriage Return, also der Code für die RETURN-Taste. Dadurch verhält sich der Computer genauso, als hätten Sie diesen String eingegeben und die Returntaste gedrückt. Die Variable A erhält deshalb den vorherigen Wert von A (127) plus 1, also 128. Nach dieser Addition wird durch GOTO 110 ein Programmneustart bewirkt. Allerdings bleiben im Gegensatz zu RUN bei einem Start mit GOTO die Variablenwerte erhalten! Drücken Sie die F1-Taste wieder, so wird der Inhalt der Variablen A weiter hochgezählt und enthält dann 129 usw. Tippen Sie das kurze Programm einfach ab, dann sehen Sie ja selbst am besten, was passiert. Entfernen Sie auch einmal den Befehl END in Zeile 140 und prüfen Sie dann, ob sich das Programm anders verhält.
Mit diesem kleinen Programm werden Ihnen in der Zeile 110 die Basic-Token gezeigt. Diese beginnen bei &80 (dez. 128) und enden bei $FF(255), allerdings sind für die Werte 254 und 255 keine Token vorhanden. Und ab 256 meldet Ihnen Ihr Computer "ILLEGAL QUANTITY ERROR", denn einen größeren Wert als 255 kann ein Byte nicht aufnehmen und muß deshalb zu einem Fehler führen. Sie können es auch einmal für die Variable A mit Werten ab 0 probieren, was dann geschieht und warum, hängt damit zusammen, daß Sie Steuerzeichen einpoken, plötzlich ist auf dem Bildschirm nichts mehr zu sehen, die Farben wechseln, es wird auf Kleinschrift umgeschaltet usw. Am meisten lernt man durch eigene Aktivitäten, also spielen Sie ruhig etwas mit dem Programm.
Wenn Sie ein paar Änderungen durchführen, könnte es Ihnen beispielsweise auf einem angeschlossenen Drucker eine Liste der Token mit den zugehörigen Werten ausgeben.

WELCHER TEXT STEHT IN WELCHER ZEILE?

Bei umfangreichen Programmen kommt es immer wieder einmal vor, daß man eine ganz bestimmte Zeile mit einem ganz bestimmten Text sucht. Oft sucht man auch einen Befehl oder

ähnliches. Mit einem kleinen Basicprogramm zeigen wir Ihnen nun, wie beispielsweise ein Programm nach ASCII-Folgen suchen kann.
Wir haben dieses Programm STRINGFINDER getauft, weil es genau das tut, was der Name sagt: Es sucht nach Zeichenfolgen und gibt bei erfolgreicher Suche die entsprechende Zeilennummer aus. Dabei benützen wir schon einige RAM-Speicherstellen, in denen der Computer sich etwas merkt, und verändern diese, um ganz bestimmte Effekte zu erreichen.

STRINGFINDER-DOKUMENTATION

Sprechen wir auch dieses Programm Zeile für Zeile durch.
In Zeile 100 steht eine reine Bemerkung, der Name des Programmes. Die Zeile 110 dient nur zur besseren optischen Gliederung und enthält nur den Doppelpunkt, dies gilt auch für die anderen Zeilen mit dem ":".
Die Zeile 120 löscht den Bildschirm und gibt den Text "STRINGFINDER – BIN BEI ZEILE:" aus. Durch Zeile 140 wird ein Zeilenvorschub ausgeführt und dann eine Eingabe angefordert. Diese Eingabe wird der Stringvariablen AS zugeordnet. Die Zeilen 170 und 180 dienen dazu, sich die alten Fensterpositionen zu merken. Bei den Computern der Serie C116 bis PLUS4 ist es ja möglich, ein Bildschirmfenster zu definieren. Die Grenzen für das Bildschirmfenster stehen in den Speicherstellen von 2021 bis 2024. Die Folge der Bildschirmkoordinaten ist: Unterer Rand, oberer Rand, linker Rand, rechter Rand. In Zeile 210 ändern wir diese Positionen durch POKEn, die Werte hierfür sehen Sie im Listing. Weshalb der Print-Befehl am Ende dieser Zeile steht, können Sie feststellen, wenn Sie in einmal weglassen: Das erste Zeichen wird noch an der alten Stelle ausgegeben.

ZUM ABTIPPEN UND ÜBEN

```
100 rem stringfinder
110 :
120 print chr$( 147)"string-finder -
 bin bei zeile: ":print
130 :
140 print:input"suchstring";a$:print
150 :
160 rem fensterpositionen merken
170 w1=peek( 2021) :w2=peek( 2022)
180 w3=peek( 2023) :w4=peek( 2024)
190 :
200 rem neues fenster definieren
210 poke 2021,20:poke2022,5:poke202
3,1:poke 2024,39:print
220 :
230 rem stringlaenge berechnen (sl)
240 rem erstes zeichen des strings
als pruefzahl (pz)
250 sl=len( a$) :pz=asc( left$( a$, 1))
260 :
270 rem bb=basic-beginn - be=basic-
ende
280 bb=peek( 43)+peek( 44)*256
290 be=peek( 45)+peek( 46)*256-4
300 bb=bb-1:rem korrektur
310 :
320 rem nz=beginn der naechsten
330 rem basiczeile im speicher
340 rem zn=aktuelle zeilen-nummer
350 nz=peek( bb+1)+peek( bb+2)*256
360 zn=peek( bb+4)*256+peek( bb+3)
370 :
380 rem alte cursorposition merken
390 p1=peek( 205) :p2=peek( 202)
400 rem cursorposition f. lfd. zeil
en-#
410 rem poken und zeilen-# ausgeben
420 poke 205,0:poke 202,30:printzn
430 rem alte cursorpositon herstell
en
440 poke 205,p1:poke 202,p2
450 :
460 for sa=bb to nz:if sa>=be then
520
470 if peek( sa)=pz then gosub 560
480 next sa
490 :
500 bb=nz-1:goto 350
510 :
```

In Zeile 250 wird die Stringlänge (SL) berechnet. Außerdem wird der ASCII-Code des ersten Zeichens als Prüfzahl für die Suche der Variablen PZ zugewiesen. Da es nicht sinnvoll wäre, den String nun im gesamten RAM-Speicherbereiches des Computers zu suchen, informieren wir den Computer über Beginn und Ende des im Speicher stehenden Programmes.
In den Speicherstellen 43 und 44 steht ein Zeiger auf den Basic-Anfang und in den Speicherstellen 45 und 46 auf den Beginn der Variablen. Dadurch können wir in den Variablen BB (Basic-Beginn) und BE (BASIC-Ende) diese Grenzen ablegen, Zeile 280 und 290.
Für den eigentlichen Programmlauf aber müssen wir dann eine kleine Korrektur durchführen, dies geschieht in Zeile 300.
In den Zeilen 350 und 360 nutzen wir das aus, was wir mittels des Monitors kennengelernt haben, nämlich die Ablage einer Basiczeile, das heißt, wir berechnen, wo die nächste Basiczeile beginnt

CURSORPOSITION WIEDERHERSTELLEN

(NZ) und die erste Zeilen-Nummer. Beim weiteren Programmlauf ändern sich diese Werte immer auf die aktuelle und die nächste Zeile.
In Zeile 390 merken wir uns in den Variablen P1 (Zeile) und P2 (Spalte) die aktuelle Cursorposition, da wir an einer anderen Stelle die laufende Zeilennummer ausgeben wollen. Diese Ausgabe soll neben dem Text erfolgen, der in Zeile 120 ausgegeben wurde. In Zeile 420 geschieht die Veränderung dieser Position und die Ausgabe der gerade aktuellen Zeilennummer. Diese Ausgabe erfolgt vor allem deshalb, damit man

```
520 poke2021,w1:poke2022,w2
530 poke2023,w3:poke2024,w4:print:p
rint
540 print"habe "fz" zeilen gefunden
!"
550 end
560 for j = sa to sa+sl
570 if peek(j)=asc(mid$(a$,j-sa+1,1
))then za=za+1
580 if za=sl then print zn,;:fz=fz+
1:return
590 next j:za=0:sa=sa+1
600 return
```

beim Programmlauf sieht, der Computer arbeitet und ist nicht abgestürzt oder hängt in einer Endlosschleife. Nachdem die Zeilennummer ausgegeben wurde, stellen wir die alte Cursorposition wieder her (Zeile 440).
Die Zeilen 460 und 470 sowie 560 bis 600 stellen das eigentliche Suchprogramm dar. Ohne ein praktisches Beispiel ist es etwas umständlich, den Suchvorgang zu beschreiben. Deshalb erklären wir ihn nur ganz allgemein.
Am Anfang ist der Wert der Variablen SA (Zeile 460) der Basicbeginn (BB). Gesucht werden soll ab der dort beginnenden Basiczeile und zwar bis zum Beginn der nächsten Zeile (NZ). Falls der Wert von SA größer oder gleich dem Programmendwert ist, soll das Programm beendet werden. Deshalb erfolgt in diesem Falle in Zeile 460 der Sprung zu 520.
In Zeile 470 wird abgefragt, ob der Speicherstelleninhalt, der im Augenblick aktuell ist, mit dem ersten Buchstaben des eingegebenen Strings identisch ist.
Falls ja, muß durch die Zeilen 560 bis 590 geprüft werden, ob auch die restlichen Zeichen übereinstimmen. Diesen Teil hätten wir zwar auch etwas anders programmieren können und dadurch eine kleine Geschwindigkeitssteigerung erreicht, aber was soll's, viele Wege führen nach Rom. Wenn Sie eine bessere Lösung haben, sind Sie eben schon etwas weiter, wir haben noch einiges auf Lager und bestimmt ist auch für Sie noch einiges dabei. Damit Sie dieses Programm aber auch in der Praxis anwenden können, wäre es sehr schön, wenn Sie es nicht jedesmal neu eingeben müssen, um in Ihrem Programm zu suchen. Selbst wenn Sie das Programm abgespeichert haben, bewirkt ein Ladevorgang, daß dadurch das bereits im Speicher stehende Programm gelöscht wird. Einen Befehl zum Verketten bzw. zur Zusammenfügung einzelner Programme zu einem Gesamtprogramm kennt Ihr Computer leider nicht.

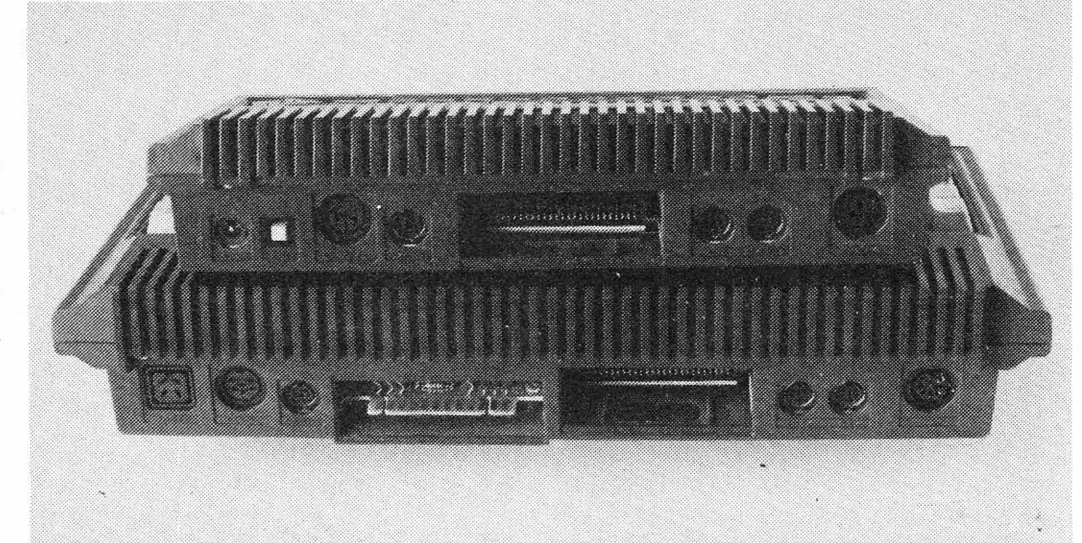
VERMISCHEN VON PROGRAMMEN

Aber auch dieses Problem meistern Sie ganz einfach, wenn Sie unsere COMMODORE WELT SPECIAL ab Seite 100 lesen, denn dort erfahren Sie, wie der Commodore dazu gebracht werden kann, zwei Programme zu einem zu vermischen. Dann können Sie wirklich mit dem Stringfinder suchen lassen. Auch dabei hilft uns der TEDMON ungemein.

(LM)

GLEICH UND DOCH NICHT GLEICH

Gleich oder nicht gleich?, das ist hier die Frage. Diese etwas freie Shakespeare Interpretation drängt sich einem auf, wenn man sich in der Computerwelt umsieht. Daß es auch bei einer Modellserie von ein und derselben Firma Unterschiede gibt, ist aus der Zeit des VC-20 schon hinlänglich bekannt. War es hier der Stromversorgungsanschluß, beim C-64 der serielle Port, so findet man beim Plus/4 wieder zwei verschiedene Stromanschlüsse. Das kann schon ganz schön Verwirrung auslösen, wenn man sich einmal nur den Computer von einem Bekannten ausleiht. Das Problem beim Plus/4 läßt sich, wie vieles bei Commodore, mit dem Herstellungsland erklären. Der obere Computer (auf dem Bild) wurde in England produziert und der untere in Deutschland. So steht es auf den Typenschildern. Daß der in Deutschland gebaute von Aldi vertrieben wurde, ist doch klar. Oder etwa nicht? Der englische wurde im deutschen Fachhandel gekauft. Alles klar?



Oben: deutscher, unten: englischer Stromanschluß

B. Welte

DFÜ FÜR BEGINNER

Eine neue, alte Dimension der Daten-Fern-Übertragung eröffnet sich den Teilnehmern mit Datex-P. Schon seit einiger Zeit gibt es diese Form der Daten-Übertragung, doch wird diese Art vorwiegend von Firmen und deren Angehörigen genutzt. Nur wenige von den sogenannten Usern nutzen die Möglichkeiten, die ihnen durch Datex-P zur Verfügung stehen. Dies soll sich nun ändern, denn die Deutsche Bundespost hat einen einheitlichen Gebührensatz für Datex-P eingeführt, zum Leidwesen der einen, oder zum Vorteil der anderen, ganz wie man das sehen will.

Doch bevor wir uns mit dem Thema Datex-P beschäftigen, sollen Sie erst einmal einen kurzen Überblick bekommen, was in der DFÜ für Abkürzungen benutzt werden und welche Bedeutung sie haben:

DFÜ-LEXIKON

ACCOUNT
Name und Paßwort des Benutzers

ANSWER/ORIGINATE
Die beiden in Verbindung getretenen Computer einigen sich anhand dieser Einstellungen auf die Träger-Ton-Frequenz. Meist ist es so, daß der Koppler bzw. das Modem des anrufenden Computers auf Stellung ORIGINATE und der Koppler bzw. das Modem der Mailbox oder des HOST auf Stellung ANSWER eingestellt ist.

ASCII
steht für American Standard Code for Information Interchange. Die Bedeutung gibt Aussage über die genormte Belegung der Bitmuster für den verwendeten Zeichensatz.

BAUD
Angabe für die Übertragungs-Geschwindigkeit (Bits pro Sekunde). In Deutschland, außer bei BTX, meist 300 Baud bei den Mailboxen und auch bei Datex-P. Wer allerdings an BTX teilnehmen möchte, der benötigt ein Modem bzw. einen Koppler mit einer Baudrate von 1200.

CARRIER
Der Trägerton, der dem Gegenüber signalisiert, daß man sende- bzw. empfangsbereit ist.

CHAT ANFORDERN
Den SYSOP rufen, um mit ihm in direkte Verbindung über die Tastatur zu treten.

DATENBITS
Zwischen fünf bis acht Datenbits werden für die Übertragung eines Bytes eingestellt.

GATEWAY
nennt man den NETZWERKRECHNER, über den man sich in andere HOSTS weiterverbinden lassen kann.

HALBDUPLEX
Die übertragenen Zeichen werden nicht geechot. Also, mit anderen Worten, werden direkt auf dem Bildschirm ausgegeben. Der Umweg über den anderen Rechner wird somit gespart, was aber nicht unbedingt ein Vorteil sein muß.

HOST
nennt man das Großrechnersystem oder die Mailbox, welche man anruft.

MODEM
MOdulator, DEModulator. Dieses Gerät zur Datenübertragung wird direkt mit der Telefonleitung verbunden. Allerdings sollte man hier unbedingt die Fernmeldegesetze in Betracht ziehen. Denn nicht jeder Computer darf mit einem MODEM ausgerüstet werden!

NETWORK
Über das NETZWERK werden mehrere Computer mit MODEM oder Kabel miteinander verbunden.

NUA
Die NETWORK-USER-ADRESS ist die Telefonnummer des Computers, der am Datex-P-Netz angeschlossen ist.

NUI
Jeder angemeldete Teilnehmer an Datex-P erhält eine Network-User-Identity, also seine Teilnehmerkennung. Diese besteht aus zwei Teilen: Teil 1 = Name und Teil 2 = Paßwort.

ONLINE
Wenn zwei Systeme miteinander in Verbindung sind.

PAD
Der Rechner der Post, über den man in das Datex-P-Netz eingespeist wird. Die Abkürzung steht für Packed Assembly Disassembly.

PARITY
Modems und Koppler geben eine Art Prüfsumme aus, womit Übertragungsfehler festgestellt werden können, je nachdem, ob die Quersumme der übertragenen Bytes 0 oder 1 ist (even or odd).

SYSOP
Der Betreiber einer Mailbox oder des Großrechners. Unter Umständen mehrere Personen, die sich verschiedene Bereiche teilen.

TIME-OUT
ist voreingestellt. Der angerufene Computer wartet nur eine gewisse Zeitspanne auf die Eingabe des Anrufenden. Sind keine Eingaben innerhalb dieser Zeit, so trennt er die Verbindung.

VOLLDUPLEX
Jedes Zeichen wird erst über die Gegenstelle geleitet und dann erst auf dem eigenen Bildschirm sichtbar.

X-MODEM
Bei Benutzung des X-Modem-Systems wird gleichzeitig noch eine Prüfsumme mit ausgesendet, welche die Aufgabe hat, die übertragenen Zeichen zu kontrollieren und eventuelle Fehler anzuzeigen.

****

Doch nun zu Datex-P:
Das Datennetz der Post, welches eine Paketvermittlung von mehreren eingebenden Com-

putern zu einem oder mehreren empfangenden Computern gewährleistet. Dieses Datennetz erreicht man über sogenannte Pads. Diese Paketvermittlung ist deswegen sinnvoll, da sie eine wesentlich geringere Zeitspanne benötigt, um Nachrichten von einem zum anderen Computer zu übermitteln. Man erspart sich so, daß man keine direkte Leitung mehr zum anderen Computer braucht.

Mit Datex-P wurde von der Deutschen Bundespost ein flächendeckendes, öffentliches Daten-Übertragungs-Netz eingerichtet, welches, technisch gesehen, auf dem Prinzip der Paketvermittlung basiert. Bei dieser Paketvermittlung geht es darum, daß abgeschickte Daten an einem Knotenpunkt gesammelt, mit anderen Daten zusammengepackt werden, sofern diese für denselben Empfänger gedacht sind, und dann weiter verschickt. Diese ganze Arbeit benötigt lediglich Bruchteile von Sekunden, woraus auch der Vorteil von Datex-P resultiert. Zusätzlich zu den ursprünglichen Daten werden noch Verwaltungsinformationen zugegeben, damit die Daten in jedem Fall auch richtig ankommen. Trotz allem wird die Reihenfolge der Daten jeweils garantiert. Untereinander sind die Netzknotenpunkte durch Leitungen mit hoher digitaler Geschwindigkeit verbunden. Eine logische Abwicklung sorgt für den Datentransfer, somit ergibt sich also, daß die Leitungen nicht mehr exklusiv für einen Anwender belegbar sind, sondern gleichzeitig mehrfach beansprucht werden können, sie werden ja jeweils nur für den tatsächlich benötigten Datenaustausch belegt. Um dieses in allgemeinen Einklang zu bringen, spricht man von einer sogenannten virtuellen Verbindung.

Was benötigt man? Tastatur, Bildschirm oder andere Anzeige, Modem bzw. Akustikkoppler.

VERBINDUNGS-AUFBAU

Man wähle die Nummer des Pad, und wenn man ein Pfeifen/Piepsen im Telefonhörer hört, aktiviere man das Modem oder den Akustikkoppler. Danach tippe man einen <.> und bestätige mit Return. Notfalls wiederhole man den Vorgang, bis sich der Pad mit <DATEX-P> und der Nummer seines Anschlusses meldet. Nun kann sich der Benutzer mit seiner NUI oder mit Angabe der NUA melden, worauf dann die weitere Verbindung gelegt wird. Nach der Eingabe der NUA bekommt der Benutzer nun die Meldung, ob die gewünschte Nummer frei ist (Verbindung hergestellt), besetzt ist (Gegenstelle belegt) oder die gewünschte Nummer nicht antwortet (Gegenstelle antwortet nicht). Sofern der Anschluß hergestellt ist, kann man seine Nachrichten, Daten eingeben.

So, das war eigentlich schon die ganze Herrlichkeit. Sollte man aber nicht im Besitz einer sogenannten NUI sein, so kann man trotzdem an Datex-P teilnehmen. Über sogenannte R-Nummern wird die Gebühr vom angewählten Rechner übernommen.

Uwe Jacob

OPTIMIERUNG VON BASIC-PROGRAMMEN

Ob Gosub oder Goto, es gibt viele Möglichkeiten, eigene Basicprogramme langsam, umständlich und verzwickt oder aber schnell, kurz und präzise lesbar zu schreiben. Was Sie wie beim Programmieren beachten sollten, erfahren Sie hier in diesem Beitrag.

Bei umfangreicheren Problemen fallen die Nachteile von BASIC gegenüber anderen Programmiersprachen unangenehm auf, besonders die langen Ausführungszeiten.

Da ein Zeitgewinn aufgrund der vorgegebenen Interpreter- oder Compilerstruktur nicht übermäßig groß sein kann, lohnt es sich, nur umfangreichere Programme nach den im folgenden aufgezeigten Gesichtspunkten zu untersuchen und ggfls. zu korrigieren. Hier lassen sich, je nach Programmqualität bis zu 15 % Laufzeitverkürzung erreichen, was jedoch auch von der Hardware abhängig ist.

1. Kommentarzeilen (REM)

REM-Zeilen, welche in das Programm geschrieben werden, verdeutlichen zwar die Struktur des Programmes und können bei der Fehlersuche bzw. Änderungen des Programmes hilfreich sein, wirken sich jedoch deutlich negativ auf die Laufzeit des Programmes aus.

Unvermeidbare REM-Zeilen sollten keinesfalls in Schleifen oder mehrfach durchlaufenden Programmzeilen stehen.

2. Leerzeichen

Blanks zwischen Befehlen oder Variablen werden zwar von den meisten BASIC-Interpretern zugelassen, können aber – besonders in Schleifen – zu Verzögerungen führen.

3. LET

Grundsätzlich gilt: je mehr Speicherplatz das

SO TEUER IST PAPIER NUN AUCH WIEDER NICHT!

Wir wissen, daß gelegentlich das Taschengeld knapp werden kann, aber trotzdem: Bitte, liebe Programmeinsender, haltet Euch bei der Beschreibung Eurer Einsendung an folgende zwei Regeln: 1. NUR 40 Anschläge (Buchstaben) pro Zeile und 2. Mindestens eineinhalbzeiliger Abstand zwischen den Zeilen, zweizeilig ist noch besser. Denn, so teuer ist ein Blatt Papier nun auch wieder nicht, daß man notfalls nicht noch ein zweites Blatt beschreiben kann.

Programm im Hauptspeicher belegt, umso größer wird die Ausführungszeit. Der LET-Befehl muß in den meisten Fällen nicht geschrieben werden. Fast alle Interpreter akzeptieren auch die Programmzeile:

10 I=I+1

anstelle 10 LET I=I+1

Ob dies zu Einsparungen zeitlicher Art führt, muß jedoch im Einzelfall überprüft werden. Es gibt Interpreter, die entgegengesetzt reagieren auf das Weglassen des LET-Befehls (z. B. BASIC LEVEL 1 von Tandy)

4. FOR NEXT Schleifen
Die Laufvariable hinter dem NEXT-Befehl sollte immer angegeben werden, sie dient der schnelleren Erkennung und Zuordnung des Schleifenendes (NEXT) zum Schleifenanfang (FOR) und damit der Beschleunigung des Programmablaufs.

Für die Ausführungszeit ist es zudem von Vorteil, wenn der NEXT-Befehl möglichst nahe dem FOR-Befehl steht, die Schleife also kompakt ist. Besondere Sorgfalt sollte daher auf die Programmschritte innerhalb der Schleifen gelegt werden. Eine durchdachte Programmierung macht sich besonders bei sehr häufig durchlaufenen Schleifen bemerkbar. Läuft eine FOR-NEXT-Schleife nach der Korrektur des Programmes auch nur 1/10 Sekunde schneller, so kann beim z. B. 1000fachen Durchlaufen der Schleife eine Zeitersparnis von 100 Sekunden erzielt werden, eine sehr lange Zeit, wenn man vor dem Bildschirm sitzt und wartet.

KOMPAKTHEIT

5. Programmzeilen
Die Zusammenfassung von Befehlszeilen — wo möglich — spart Speicherplatz und Rechenzeit. Mehrere Befehle in einer Zeile werden schneller bearbeitet als mehrere Zeilen mit jeweils nur einem Befehl. Es muß aber gewarnt werden vor vollgepackten Zeilen. Es gibt gerade beim Commodore-BASIC verschiedene Tricks eine übermäßig große Anzahl von Einzelbefehlen in eine Zeile zu pressen (mit Abkürzungen o. ä.), dies könnte sich u. U. negativ auf die Ausführungszeit auswirken.

ES GEHT AUCH ANDERS

Beispiel einer möglichen Programmoptimierung:

```
5 FOR I = 1 TO 25
10 LET A = B + I
15 PRINT A
20 NEXT
```

Schneller läuft diese Programmzeile in der Form:

```
5 FOR I = 1 TO 25
: LET A = B + I
: PRINT A : NEXT I
```

Schneller wird es noch so: (ohne LET und ohne Blanks)

```
5 FORI=1TO25:PRINT
(B+I):NEXTI
```

Hier wurde die LET-Zuweisung komplett weggelassen und die Berechnung des Wertes in den PRINT-Befehl gepackt. Außerdem wird Speicherplatz, welcher für die Variable A benötigt würde, gespart.

6. Formeln
Der Aufbau mathematischer Formeln hat entscheidenden Einfluß auf die Rechengeschwindigkeit. Divisionen nehmen im allgemeinen mehr Zeit in Anspruch als Multiplikationen.

```
10 A = B / .001
```

Das gleiche Ergebnis erhält man schneller mit der Zeile

```
10 A = B * 1000
```

Wann immer es möglich ist, sollte mit ganzen Zahlen gerechnet werden, denn diese werden schneller bearbeitet als Zahlen, die gebrochen sind. Dies gilt übrigens für alle Programmiersprachen und nicht nur für BASIC.

7. GOTO und GOSUB
Ein großer Nachteil von BASIC besteht darin, daß ein strukturiertes Programmieren nicht unterstützt wird. Dies kann sich negativ auf die Ausführungszeiten von Programmen auswirken. Der BASIC-Interpreter läßt GOTO-Sprünge in nahezu beliebiger Form zu. Man sollte aber bedenken, daß GOTO-Sprünge über große Entfernungen im Programm zum einen dieses unübersichtlich machen, zum anderen Zeit kosten. Ebenso verursachen GOSUB-Befehle Verzögerungen. Jedoch ist der Vorteil eines mehrfach genutzten Unterprogrammes mit dem Zeitfaktor abzuwägen.

8. Rechengenauigkeit
Beim Erstellen eines Programmes sollte vorher klar sein, welche Genauigkeit der Ergebnisse verlangt wird. Oft verleitet die mögliche Genauigkeit der Computer eine Berechnung von vielen Nachkommastellen, obschon dies in manchen Fällen nicht notwendig ist.

Höhere Genauigkeit = höherer Rechenaufwand = längere Ausführungszeit.

9. Speicherkapazität
Ein längeres Programm läuft weniger schnell als ein kurzes — logisch. Extreme Laufzeitverlängerungen muß man jedoch in Kauf nehmen, wenn die Speicherkapazität des Rechners bis an die Grenze der zur Verfügung stehenden Hauptspeicherkapazität ausgenutzt wird.

Kurz vor dem Erreichen der Kapazitätsgrenze steigt die Ausführungszeit eines Programmes dramatisch. Es ist daher zu überlegen, ob ein komplexes Problem in mehrere Einzelschritte zerlegt werden kann, in Form verschiedener kleiner Programme, die nicht gleichzeitig im Hauptspeicher abgelegt werden müssen.

IN DER KÜRZE LIEGT DIE WÜRZE

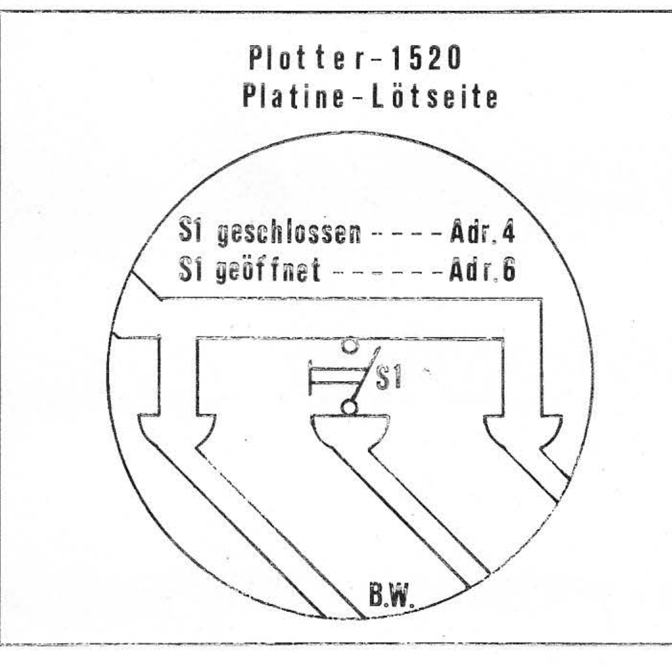
Zusammenfassend können folgende Regeln im allgemeinen eine Laufzeitverkürzung von BASIC-Programmen bewirken:

- keine REM-Zeilen verwenden
- keine unnötigen Blanks schreiben
- LET-Befehle weglassen
- Laufvariablen hinter NEXT immer angeben
- Zeilenanzahl reduzieren durch Straffung des Programmes
- mathematische Formeln überarbeiten
- GOTO-Sprünge über große Entfernungen vermeiden
- GOSUB-Befehle möglichst vermeiden
- FOR-NEXT-Schleifen möglichst kurz halten
- keine übersteigerte Genauigkeit ansetzen
- Kapazitätsgrenzen der Anlage nicht völlig ausnutzen

Wie schon erwähnt, reagieren nicht alle BASIC-Interpreter gleich, und es muß im Einzelfall ausprobiert werden, welche Maßnahmen das Programm optimieren.

Sekundäradresse Umschalter, am Commodore 1520 Plotter

DER COMMODORE-PLOTTER KANN AUCH DRUCKEN!



Wer hätte es sich nicht schon gewünscht: Mit dem Plotter auch drucken. COMMODORE WELT zeigt Ihnen, daß das anscheinend Unvereinbare — zwei verschiedene Adressen werden benötigt — miteinander zu koppeln, funktioniert

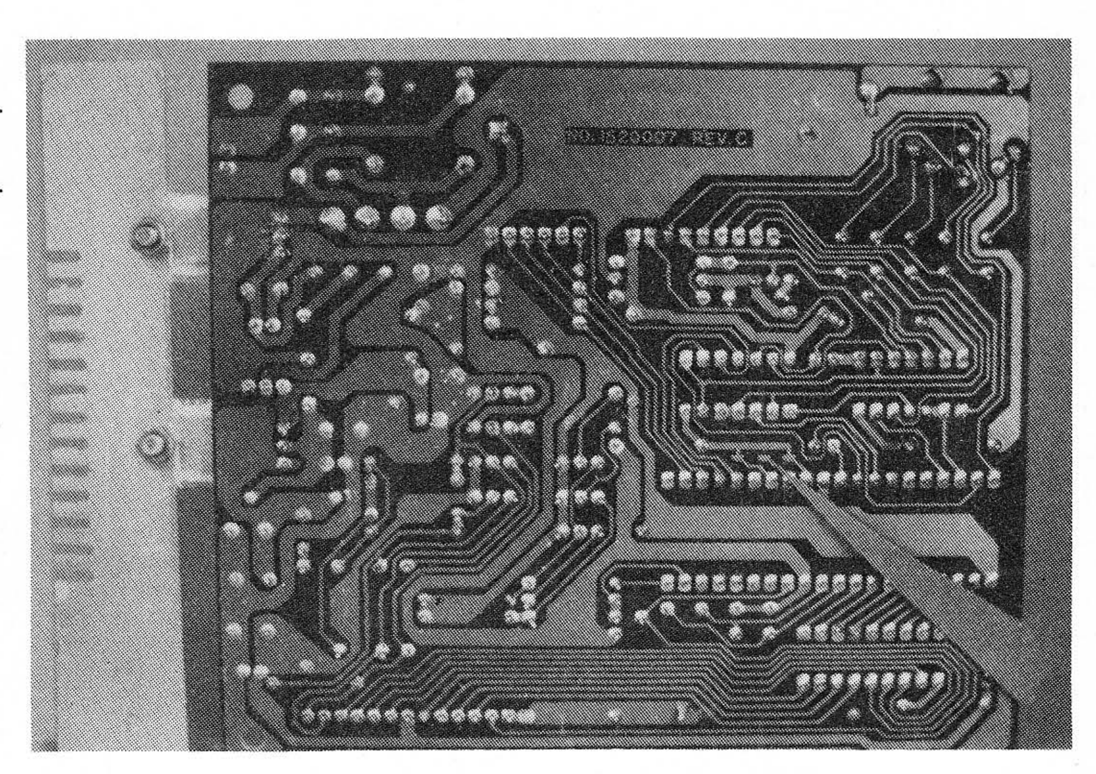
Der Plotter 1520 von Commodore hat, obwohl durch seine Papierbreite beschränkt, doch eine Menge auf dem Kasten. Auch sein zur Zeit günstiger Preis macht ihn für viele Anwender erstrebenswert. Man kann mit ihm zeichnen, schreiben und sogar eine Hardcopy anfertigen. Doch die Sache hat einen Schönheitsfehler. Die meisten Programme, die auf Drucker und dergleichen zugreifen, benutzen nämlich die Sekundäradresse <4>. Der Plotter 1520 ist aber von Haus aus auf die Adresse <6> eingestellt. Softwarelösungen zur Adressenänderung, wie bei der Diskettenstation, schlagen



hier fehl. Der Plotter weigert sich beharrlich, eine neue Adresse anzunehmen. Da bleibt einem nur noch die Möglichkeit, die Sache hardwaremäßig anzugehen. Und siehe da!, Commodore hat diese Möglichkeit vorgesehen. Auf der Platine des Plotters befinden sich sogenannte ‚Jumper', mit denen die Adresse geändert werden kann. Doch dazu etwas später. Zuerst muß der Plotter geöffnet werden. Dazu werden die zwei Befestigungsschrauben gelöst und das Oberteil nach hinten geklappt. Jetzt erkennt man drei Kabel, die die Platine mit den Tasten auf der Oberseite, der Plottermechanik und der Stromversorgung verbinden. Diese müssen nun gelöst werden. Der vordere Stecker, der zum eigentlichem Plotter führt, hat keine eindeutig zu identifizierende Aufstecksstellung und sollte daher gekennzeichnet werden. Bei den beiden anderen ist dies durch die Rastzungen nicht nötig. Anschließend werden die Befestigungsschrauben, die die Platine mit dem Boden verbinden, gelöst. Auch der Kühlkörper wird mit der Platine zusammen entfernt. Auf keinen Fall die beiden Schrauben lösen, mit denen die Spannungs-

PLATINE DREHEN

regler auf dem Kühlblech befestigt sind! Nach dem Entfernen der Schrauben kann die Platine herausgenommen werden. Um an die Jumper zu kommen, muß die Platine gedreht werden, so daß die Lötseite oben zu liegen kommt. Bild (1). Die ‚Jumper' befinden sich auf der rechten Seite im mittleren Teil der Platine. Bild (2) farbiger Pfeil. Wird nun zwischen dem mittleren ‚Jumper' und der darüber befindlichen Leiterbahn eine Brücke hergestellt, wird der Plotter auf die Adresse <4> eingestellt. Um die Sache etwas flexi-



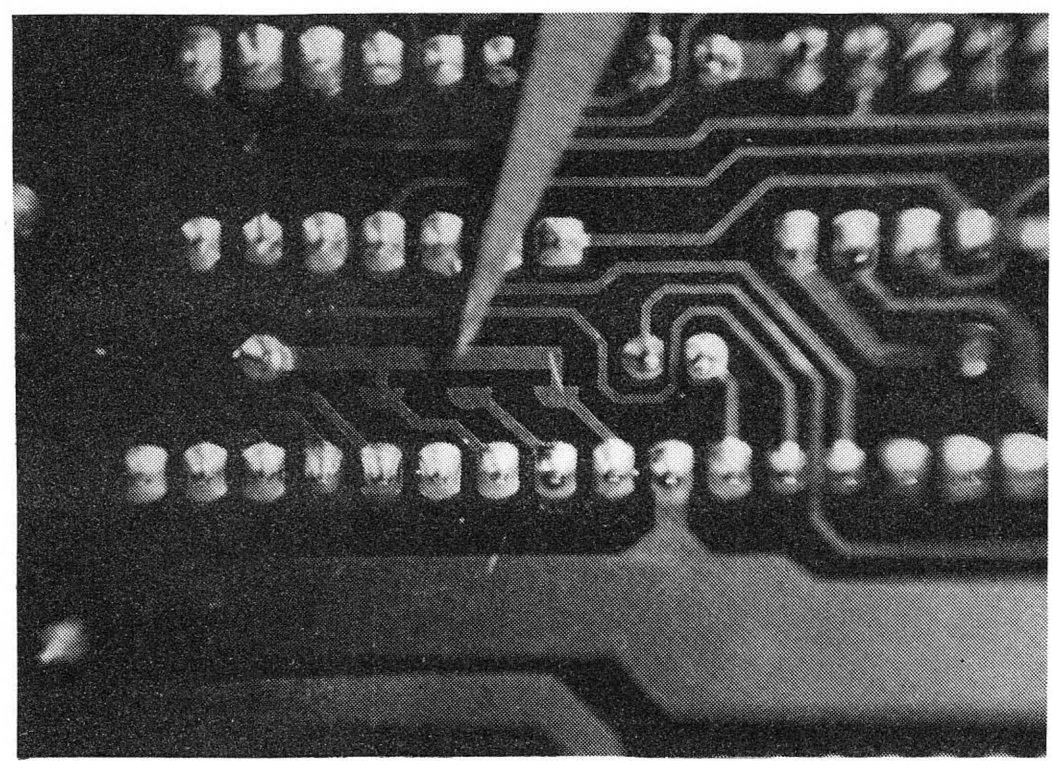
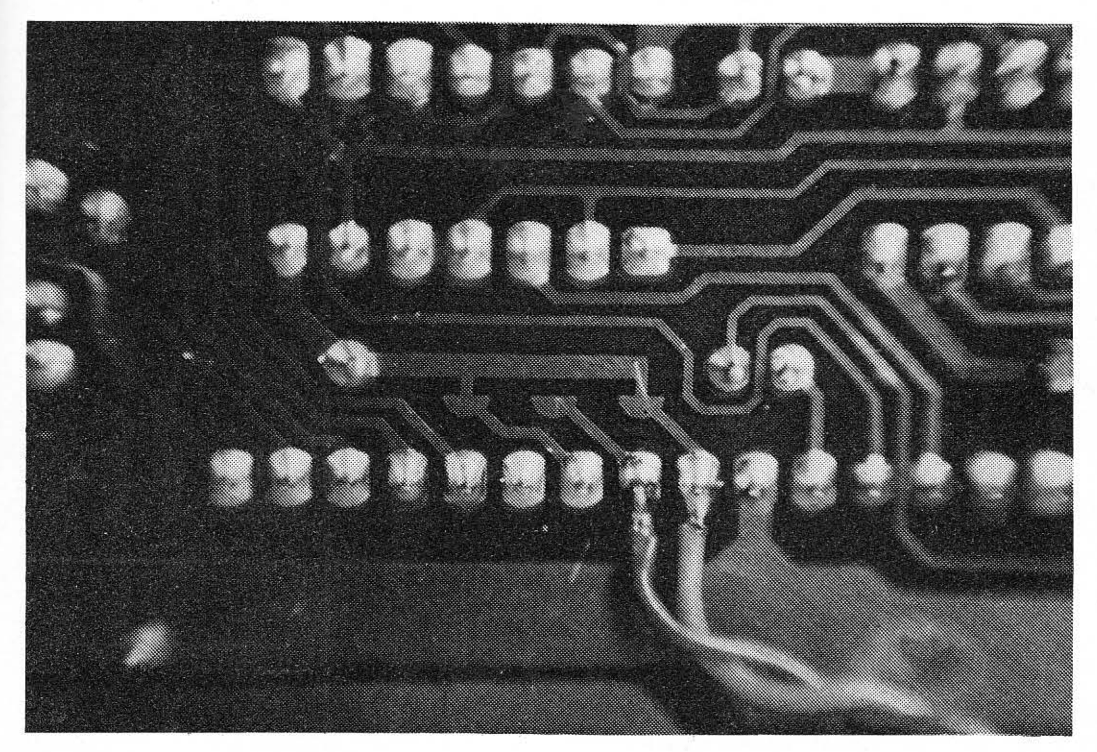


Foto oben: Die Jumper auf der Lötseite der Platine. Foto unten: Hier muß eine Brücke in die Platine eingelötet werden.



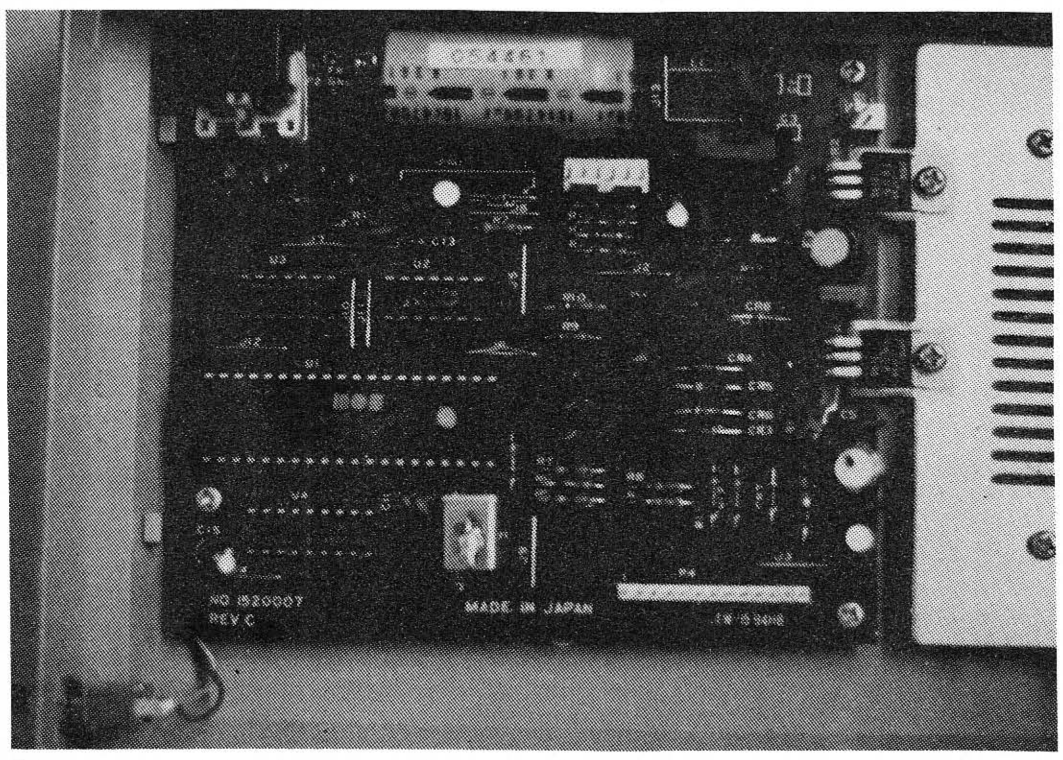


Foto oben: Eingelötete Verbindungsleitungen zum Umschalter. Foto unten: Bestückungsseite mit Digitalschalter links unten

bler zu gestalten, bietet sich der Einbau eines Schalters an, mit dem dann zwischen Adresse <4> und <6> hin und hergeschaltet werden kann. Durch diese Möglichkeit können alle Programme, die die Adresse <6> benutzen, weiter verwendet werden. Durch Umschalten auf die Adresse <4> werden jetzt auch Textprogramme, die keine Möglichkeit der Adressenänderung haben, plotterfähig. Sogar eine Bildschirmhardcopy mit 'SIMONS-BASIC' ist jetzt möglich. Ändert man z.B. beim Plotter die Zeichengröße auf 80 Zeichen, kann jetzt ein Dokument in voller Breite, inclusive aller Formatbefehle (einrücken, linker Rand, rechter Rand, Blocksatz usw.), ausgedruckt werden. Nur mit den Umlauten hat er so seine Probleme. Um den Plotter auf 80 Zei-

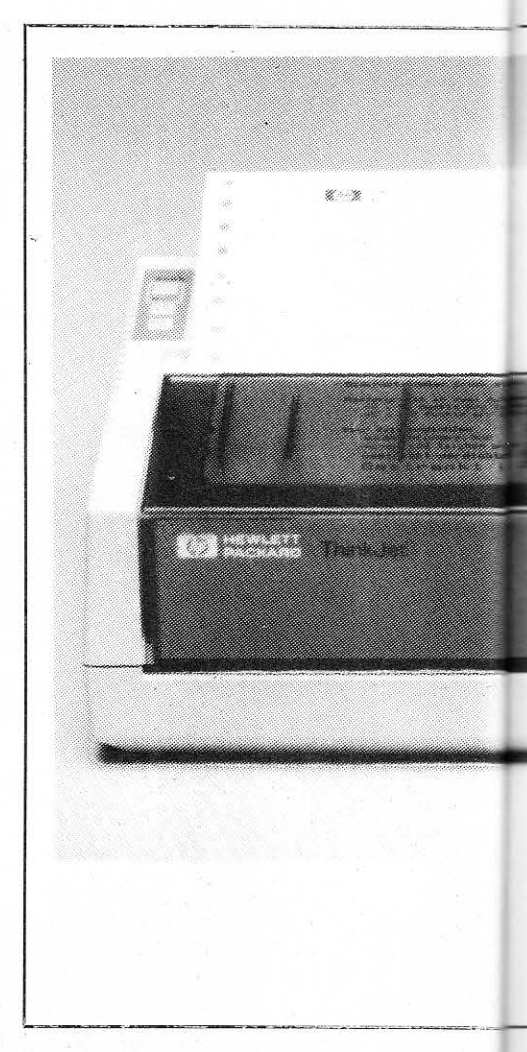
ARBEITET PRIMA

chen einstellen zu können, muß vor dem Start des Textprogrammes ein entsprechender Befehl zum Plotter gesendet werden. Dieser Befehl wird im Direktmodus eingegeben und lautet folgendermaßen. OPEN3,4,3:PRINT #3,X. Für X wird die Schriftgröße eingegeben und zwar wie im Handbuch auf Seite 27 beschrieben. 0=80, 1=40, 2=20, 3=10 Zeichen pro Zeile. Nach der Eingabe kann dann das Textprogramm gestartet werden. EASY SCRIPT auf dem C 64 und SCRIPT/PLUS auf dem Plus/4 arbeiten tadellos mit dem Plotter zusammen. Allerdings muß man auf einige Möglichkeiten der Matrixdrucker, wie Groß, Fettschrift und andere Sachen verzichten. Aber für einen Kontrollausdruck, Notizen und dergleichen, dann aber in Schönschrift, reicht es allemal.

B. Welte

Druckerwissen :

INKJET: NEUE DIMENSIONEN FÜR DEN COMPUTER-ANWENDER?



Viele winken nicht nur beim Wort „Nadeldrucker" entsetzt ab, sondern beim Begriff „Matrix" überhaupt. Das kommt eben – wie bereits erklärt – daher, daß dieser Druckertyp sein lärmendes und von der Schrift her unschönes „Vorleben" hat. Dabei gibt es schon seit ein paar Jahren Matrixdrucker, die gelegentlich als „King der Computer-Printer" bezeichnet wurden. Vielleicht auch deshalb, weil man vor zwei Jahren noch mindestens 10 000 Mark bezahlen mußte, um mit seinem Computer-Drucker in den begehrten Jet-Set zu gelangen. Der „Jet", das ist nämlich der „Strahl". Beim Flugzeug wie beim Drucker. Und ein Ink-Jet ist eben ein Tintenstrahl.

Auch der Tintenstrahl – alias Inkjet-Drucker ist ein sogenannter Matrixdrucker. Diese Eigenschaft hat er mit dem Nadeldrucker gemeinsam. Er kann alles, was der Nadeldrucker kann, aber er ist schneller und präziser.

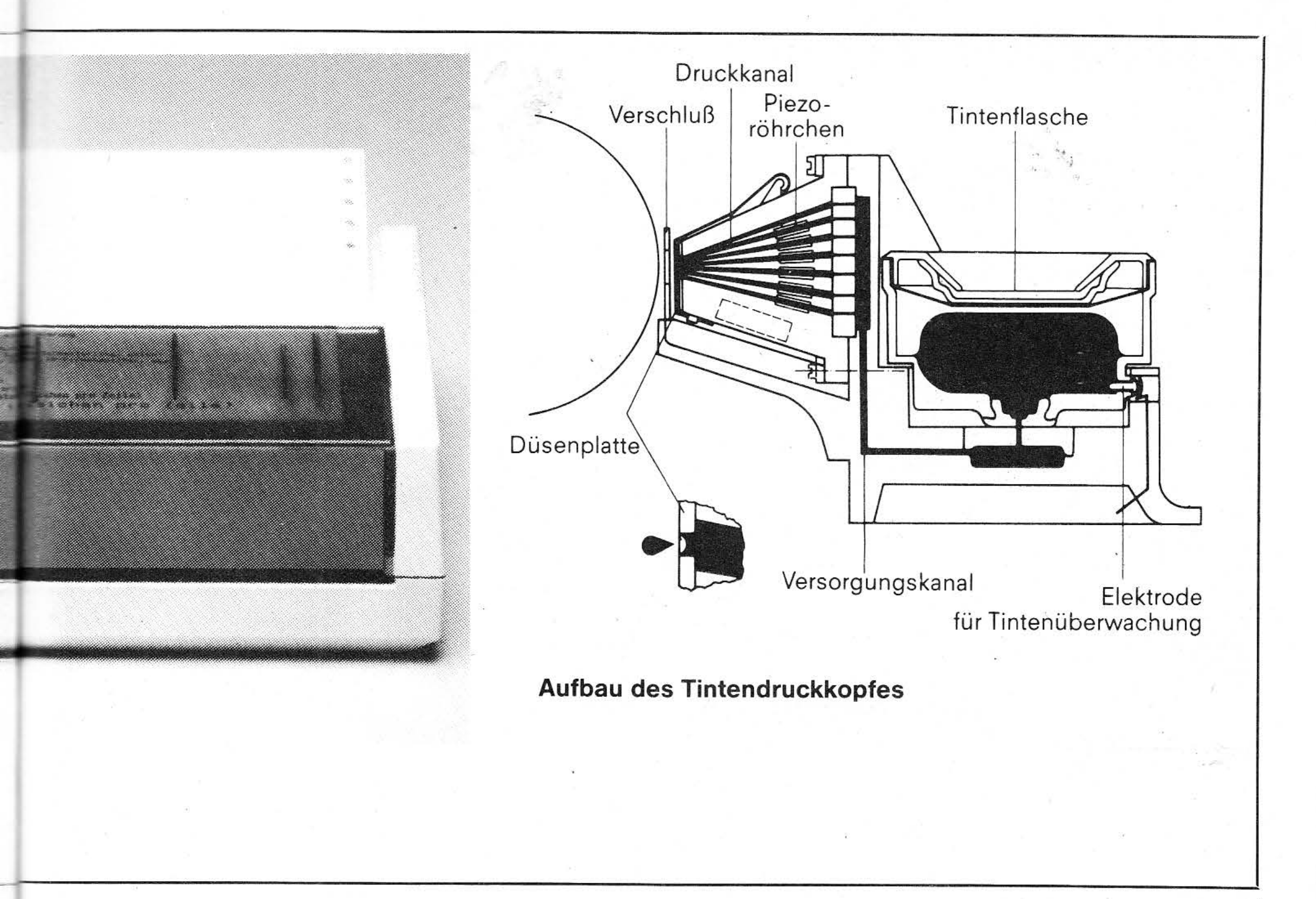
In beiden Fällen werden die darzustellenden Zeichen aus Einzelpunkten zusammengesetzt. Stellt man – um es aus dem Kapitel über Nadeldrucker zu wiederholen – sich eine 5x7 Matrix vor, so ist das ein Muster, das aus sieben senkrecht angeordneten Punkten besteht, von denen jeder in fünf waagerechten Positionen drucken kann. Betrachten Sie sich noch einmal die betreffenden Abbildungen im Abschnitt „Nadeldrucker".

Nun ist Tinte kein Stahlstift, sondern eine Flüssigkeit. Entsprechend „beweglich". Man kann sie unter anderem zerstäuben. Genau das macht nämlich der Tintenstrahldrucker. Während heute bei den Nadeldruckern die 7x7- oder 9x9-Matrix der Standard ist, sieben oder neun Nadeln im Druckkopf untergebracht werden können, lassen sich beim Tintenstrahldrucker bis zu 30 Düsen auf dem gleichen Raum unterbringen. Einfach deshalb, weil bei Zerstäuberdüsen eine andere Präzision möglich ist als bei Stahlstiften. Die lassen sich nicht in Dimensionen von 1/1000 Millimeter und feiner anfertigen und handhaben – schon ein nicht glattliegendes Papier würde eine solche „Nadel" zerstören. Bei Zerstäuberdüsen ist es möglich. Dadurch ergibt sich ein „Punktmuster", das als solches nicht mehr zu erkennen ist.

Tinte ist beweglicher

Jeder, der jemals eine Sprühdose verwendete (bitte, jetzt nicht mehr tun – diese chemischen Keulen sind Umweltkiller), weiß, daß eine Flüssigkeit in einem Sekundenbruchteil in Milliarden kleinster Teilchen zerstäubt werden kann.

Das tut der Inkjet-Drucker. Das Ergebnis ist einerseits ein scharfes Schriftbild, von dem eines Typenraddruckers kaum zu unterscheiden. Andererseits sind diese Drucker dank der physikalischen Eigenschaften des Farbüberträgers, nämlich der Tinte, sehr schnell. Weshalb der Inkjet-Drucker zur ganz schnellen Truppe gehört. Bis zu fast 700 Zeichen pro Sekunde bringt er zu Papier.



Aufbau des Tintendruckkopfes

Allerdings, und da beginnen die Nachteile, nicht auf jedem Papier. In den meisten Fällen ist ein besonders saugfähiges Spezialpapier notwendig – die Tinte darf sich darauf gar nicht erst „ausbreiten" – nicht mal um winzigste Millimeterbruchteile, weil es sonst nur Klecksreihen statt lesbarer Zeichen gäbe. Neuerdings gibt es auch Tintenstrahldrucker, mit denen auf normalem satiniertem Papier gedruckt werden kann, dem gleichen Papier, das auch die Thermo-Transfer-Drukker verlangen und bei dem der Begriff „normal" doch sehr geschmeichelt ist. Man bekommt es allenfalls in wirklich großen und gut sortierten Bürobedarfsgeschäften und eventuell bei Druckereibedarfshandlungen, die aber an Direktverbraucher meist nicht verkaufen.

Nun, für den „normalen" Anwender sind Tintenstrahldrucker nicht gedacht, wenn auch ihre Preise deutlich nachgegeben haben. Man kann nun schon mit 2000 bis 3000 Mark dabei sein, wobei man allerdings doch erhebliche Abstriche bei der Geschwindigkeit machen muß. Die Spitze liegt in dieser

Für „Normal-Anwender" nicht gedacht

Preisklasse bei 150 cps, die meisten Inkjets schaffen nur 20 bis 80 cps – da ist man von der Qualität her beim Typenraddrucker besser aufgehoben. Will man die beste Tintenstrahlqualität und dazu auch die hohe Geschwindigkeit bis zu 400 cps haben, so muß man immer noch um die 8000 Mark ausgeben. Die Höchstgeschwindigkeit von 600 und mehr Zeichen pro Sekunde ist vom Preis her für den Anwender mit dem Personal-Computer unerschwinglich; er liegt in der Preisklasse, mit der man bei Groß-EDV-Anlagen rechnen kann.

Daß Inkjet-Drucker nach zwei verschiedenen Methoden arbeiten, nämlich entweder nach dem Unterdruck- oder dem Dampfblasenprinzip, ist unter diesen Umständen nicht erörterungsbedürftig.

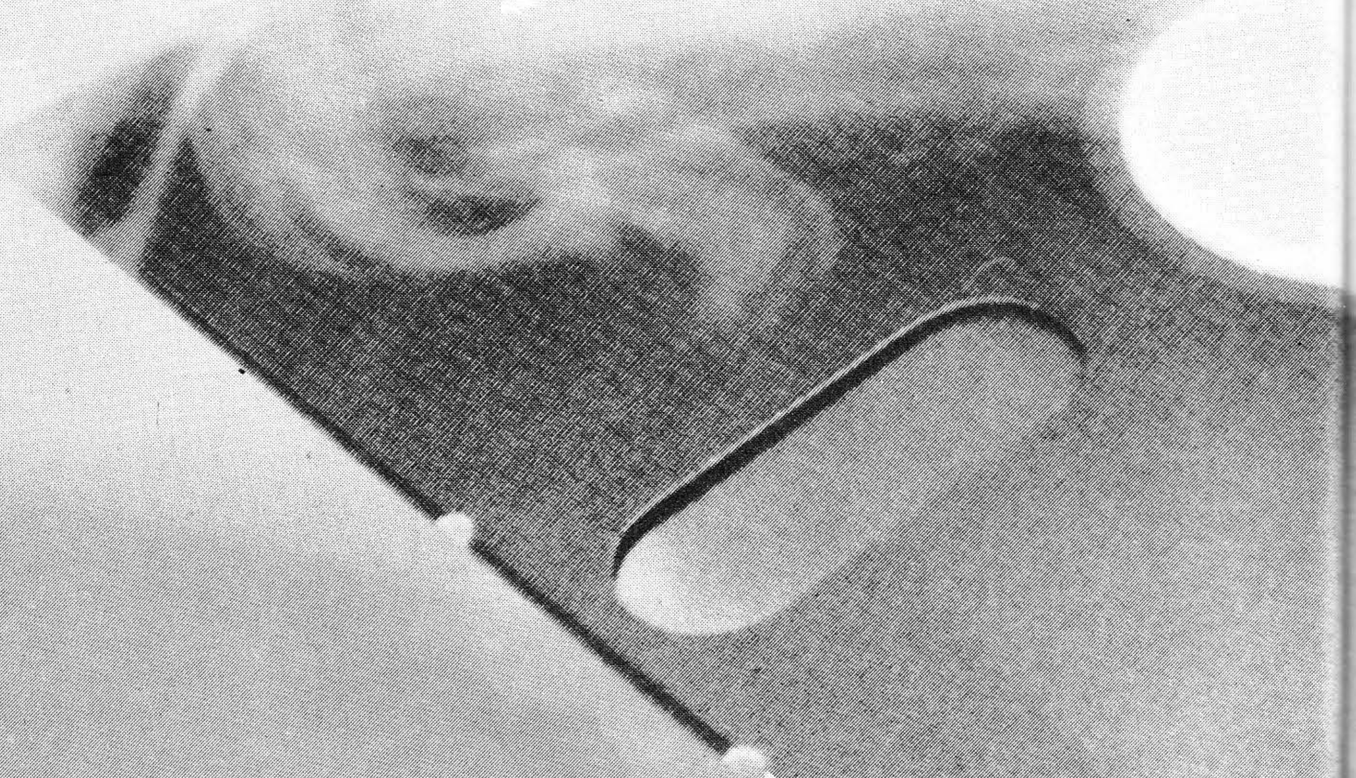
Das Angebot ist auch sehr klein – offenbar trauen die Anbieter dieser Technik keinen breiten Durchbruch zu.

Sie ist zudem bereits überholt. Das, was der Tintenstrahldrucker besser kann als der Nadeldrucker und was er mehr kann als der Typenraddrucker, haben neue Techniken inzwischen eingeholt: Farbfähigkeit, Lautlosigkeit, bestes Schriftbild und dazu Graphikfähigkeit, schließlich die Fähigkeit, auf Folien zu schreiben.

Seine „Blütezeit" hat der „Jet" unter den Druckern nie erlebt. In der oberen Leistungsklasse ist er – der erforderlichen komplizierten Technik und Elektronik wegen – nach wie vor für PC-Anwender zu teuer. In der Normalklasse aber hat er viel Konkurrenz in neueren Techniken.

Für Technik-Feinschmecker oder für Technik-Verliebte ist er allerdings eine Delikatesse.

DISKETTE ENTMYSTIFIZIERT



DIE WESENTLICHEN MERKMALE

Wenn Sie kein Computer-Fachmann sind, wollen Sie sich nicht mit einer Menge unnötiger technischer Einzelheiten belasten. Aber Sie möchten sicher sein, für Ihren eigenen Computer die richtigen Disketten zu wählen und diese auch so zu benutzen, daß die Datensicherheit gewährleistet ist.
Für den Nichtfachmann ist die Wahl der richtigen Diskette äußerst einfach – sie hängt ganz und gar von der Bauart und dem Modell Ihres Computers ab.
In der Computer-Fachliteratur haben Sie wahrscheinlich eine Vielzahl verwirrender Ausdrücke gelesen. Nachstehend geben wir Ihnen eine vereinfachte Erklärung der hauptsächlichen Begriffe.

Magnetspeicher
Eine Platte oder Band mit Spezial-Oxidbeschichtung, die in der Lage ist, magnetische Informationen zu speichern.

Platte, Diskette, Flexible Disk, Floppy
Diese Ausdrücke haben alle die gleiche Bedeutung – eine 8"-, 5 1/4"- oder 3 1/2"-Platte aus elastischem Kunststoff, die speziell in Computer-Plattenlaufwerken verwendet wird. Die meisten Personal Computer verwenden entweder 5 1/4"-Platten oder die neuen 3 1/2"-Platten. ("=Zoll)

Miniplatte, Mikroplatte
Diese Begriffe beziehen sich auf 5 1/4"- bzw. 3 1/2"-Platten.

Einfache -, doppelte -, vierfache Dichte
Diese Begriffe bezeichnen die Menge der Informationen, die auf einer Plattenseite gespeichert werden kann; die von Ihnen verwendete Dichte hängt von Ihrem verwendeten Computer-, Diskmodul ab.

Einseitig, doppelseitig
Gewisse Computer verwenden beide Seiten einer Platte, andere jedoch nur eine Seite. Sie können eine doppelseitige Platte, wenn die Schreibdichte stimmt, in einem einseitigen Plattenlaufwerk einsetzen. Aber es besteht kein Grund dafür, da die Maschine nur eine Seite liest und beschreibt. Dagegen können Sie eine einseitige Platte nicht unbedingt in einem doppelseitigen Laufwerk verwenden, weil die Rückseite meistens nicht getestet wurde und folglich keine Fehlerfreiheit von den Diskherstellern garantiert wird.

Spuren
Die Daten werden vom Plattenlaufwerk automatisch auf getrennte konzentrische Spuren aufgezeichnet. Gewisse Plattenlaufwerke beschreiben 40 Spuren pro Seite, andere 77 oder 80 Spuren. Dies hängt wiederum von Ihrer verwendeten Maschinenkonfiguration ab.

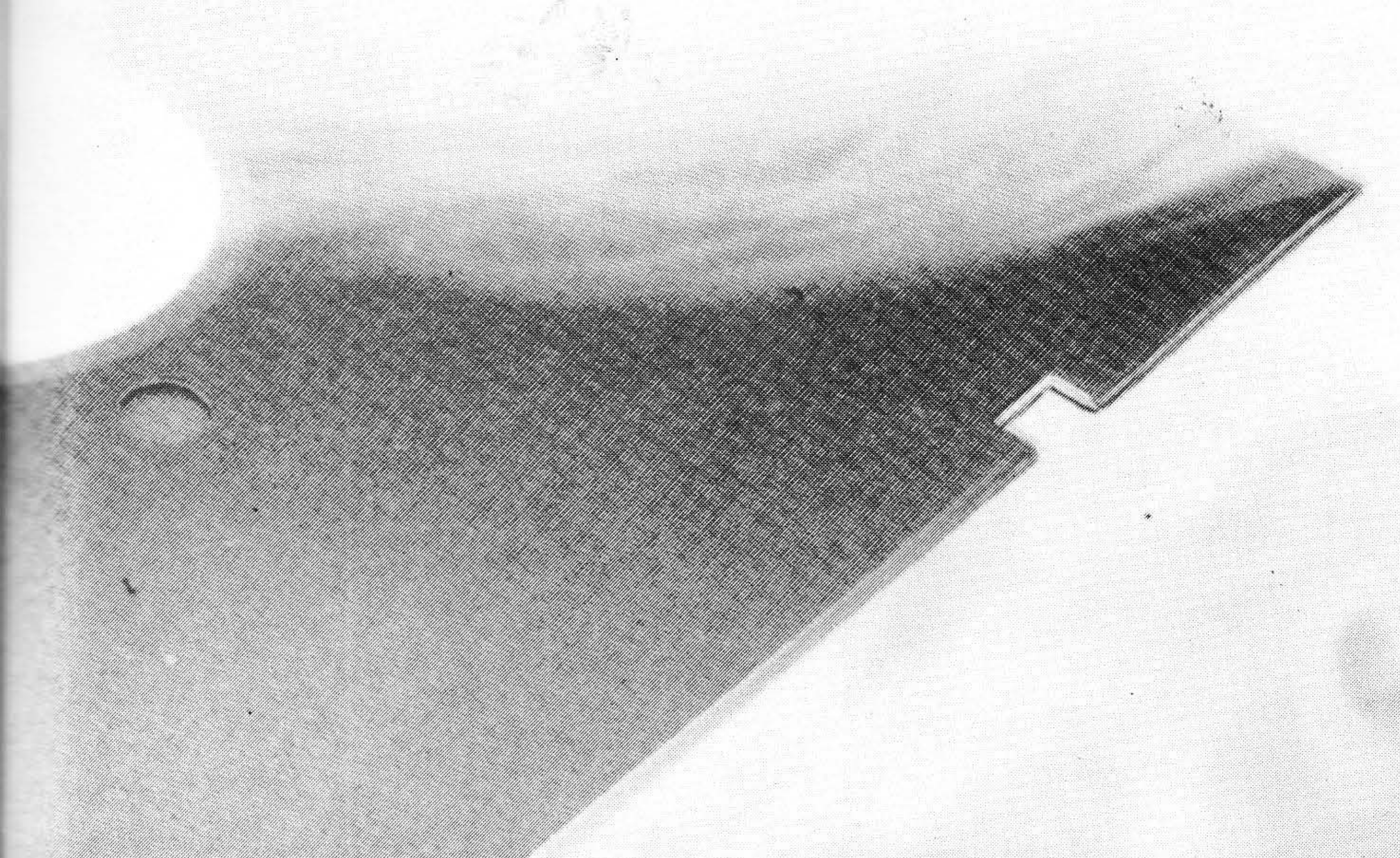
Indexloch
Ein Bezugsloch, das der Maschine als Richtpunkt dient, um die Platte in einzelne Sektoren zu teilen.

Sektor
Die Platten sind in Sektoren unterteilt, so daß das Betriebsprogramm der Maschine Informationen genau auf jeder Spur auffinden kann.

Hard Sectored
Eine Platte, die in eine bestimmte Anzahl Sektoren unterteilt ist, wobei diese Sektoren durch kleine Löcher in der Platte begrenzt sind (zusätzlich zu dem Indexloch). Z.B. zeigen 10 Löcher eine Platte mit 10 Sektoren an.

Soft Sectored
Eine Platte, die (während des Formatierens) von der Maschine automatisch in Sektoren eingeteilt wird und nur ein Indexloch besitzt.

Technische Daten entnommen aus einer Data Life Broschüre der Firma VERBATIM GmbH Eschborn.



WIE FUNKTIONIERT EINE DISKETTE?

Wenn Sie eine Diskette in das Laufwerk einlegen und die Verriegelung schließen, beginnt sie sich zu drehen. Die normale Geschwindigkeit ist 300 Umdrehungen pro Minute. Der Lese-/Schreibkopf bewegt sich radial über die Plattenoberfläche (innerhalb des Bereichs, der durch die längliche Öffnung in der Plattenhülle begrenzt ist). Das Maschinenbetriebssystem weist den Kopf an, wo er zu lesen oder zu schreiben hat, und zwar nach seiner eigenen Logik.
Ein großer Vorteil der Platten im Vergleich zu Magnetbändern besteht darin, daß die Daten nicht sequentiell bearbeitet werden brauchen. Der Lese-/Schreibkopf geht direkt zu der Stelle auf der Platte, die im Inhaltsverzeichnis automatisch angelegt wird, wenn Sie Daten in die Maschine eingeben.

WIE SIE IHRE DISKS BEHANDELN SOLLTEN!

Einwandfreie Leistungen machen den engen Kontakt zwischen Lese-/Schreibkopf und Plattenoberfläche erforderlich. Alles, was diesen Kontakt stört, kann mit Datenverlust enden. Daher sind einige Dinge zu beachten, um das Überleben Ihrer Daten sicherzustellen. Diskette sorgfältig ausrichten und einlegen, (ebenfalls zur Sicherheit Ihres Systems). Disketten bei Nichtgebrauch in ihren Schutzhüllen aufbewahren. Disketten zweckmäßigerweise bei einer Temperatur zwischen 10 und 50 Grad Celsius aufbewahren.
Disketten nicht biegen und falten. Disketten nicht schädlichen Magnetfeldern aussetzen (die durch Leuchtstofflampen, Ventilatoren, Staubsauger, Lautsprecher oder sonstige Elektrogeräte entstehen). Ein Mindestabstand von 60 cm ist angemessen. Keine Büroklammern oder Gummibänder im Zusammenhang mit den Disketten anwenden. Nicht mit Kugelschreiber, Stiften oder gar Bleistiften auf die Disketten schreiben, und nicht versuchen, aufgeklebte Etiketten abzukratzen. Nichts auf die Disketten legen. Möglichst stehend in Platikkästen aufbewahren. Die Aufnahmeseite nicht berühren; diese vor Krümeln oder verschütteten Flüssigkeiten schützen.

5 WICHTIGE VORTEILE, DIE DAS LEBEN IHRER DATEN SCHÜTZEN

Nabenringe
Die Disketten sollten mit das Mitteloch verstärkenden Nabenringen versehen sein, die bei der Zentrierung der Diskette nützlich sind, die Abnutzung im Nabenbereich herabsetzen und eine bessere Ausrichtung ermöglichen.

Die Hülle
Die Diskettenhülle sollte eine gewisse Stabilität besitzen. Der innere Filz sollte die Platte von Schmutz befreien und unnötige Abnutzung vermeiden sowie den Abrieb einschränken.

Das Gleitmittel
Es sollte ein Gleitmittel verwendet durch Polieren eine gleichmäßige Glätte besitzen. Je glatter die Platte, desto besser schützt sie gegen Abnutzung des Lese-/Schreibkopfes.

Prüfstandards
Die Diskette sollte im mindesten den Normvorschriften entsprechen oder sogar noch darüber liegen. Ausgangsmaterial und Beschichtungsrezepturen in ihrer chemischen, magnetischen Zusammensetzung sollten eine einwandfreie Verwendung sicherstellen.

FORDERUNGEN DER DISKETTEN-ANWENDER

A) Preiswürdigkeit
B) Datensicherheit
C) Qualitätsstandard
D) Produktkennzeichnung
E) Robustheit

Bauvorschlag:

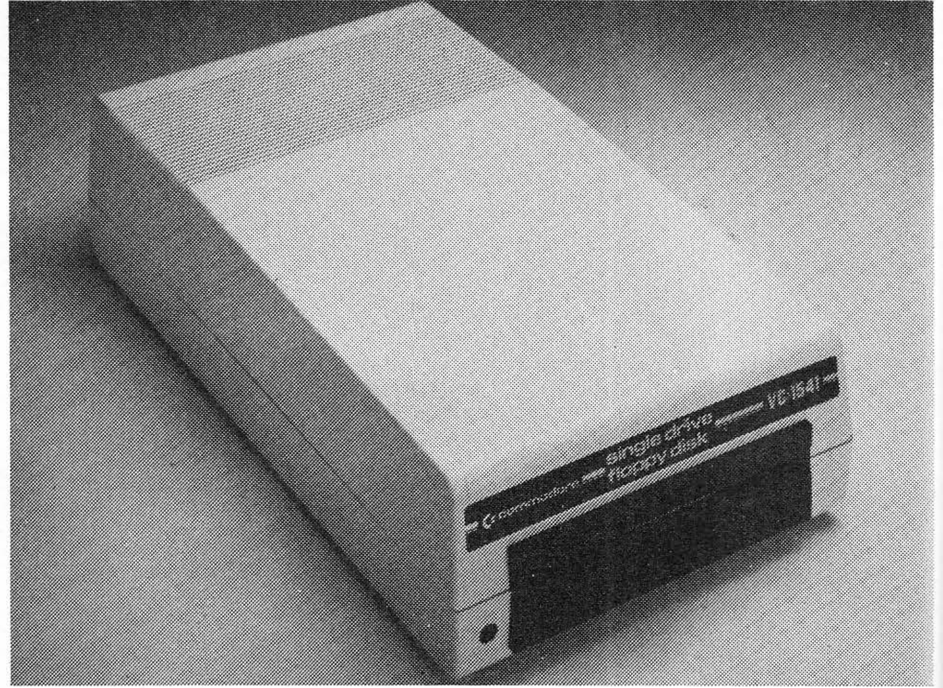
DOPPELFLOPPY

Jeder, der sich eine zweite Floppy zulegen möchte, kennt den Vorteil einer Doppelfloppy (z.B. 4040): Den wahlfreien Zugriff auf die beiden Laufwerke. Will man nun aber eine zweite 1541 anschließen, funktioniert das einfach nicht. Der Grund: Die zweite Floppy ist nicht Laufwerk Nr. 1 des Geräts Nr. 8, sondern Laufwerk 0 von Nr. 9. Will man dieses ansprechen, so kann man dies nicht im Filenamen ("1: Name") machen. (Bei einem Versuch meldet die Floppy 8 einen "Drive not ready"-Fehler, denn es ist gar nicht vorgesehen, ein zweites Laufwerk anzuschließen.) Vielmehr muß die Selektion bereits im OPEN-Befehl geschehen, durch die Angabe der Geräteadresse 9. Hier liegt das Problem: Denn welches Programm erlaubt diese Option?

Schauen wir uns einmal an, welche Möglichkeiten sich ergeben, auf zwei Floppys wahlweise zuzugreifen: Zum einen wäre ein Anschluß eines reinen zweiten Laufwerks (ohne Steuerplatine) bei gleichzeitiger Änderung des Betriebssystems denkbar. Denn das 1541-ROM ist nur eine gestutzte Version des 4040-ROM. (Sehen Sie sich es bei $F2CD bis $F2D7 an: Hier wird die Angabe "1:" durch Ausgabe des Fehlers 74 abgewürgt.) Nun wäre dies sehr aufwendig und jeder Umbau in der Floppy geht auf Kosten der Kompatibilität. Eine andere Möglichkeit wäre eine Lösung im Computer (wie z.B. Supertwin). Hier wird ein "1:" gleich im Computer abgefangen und als "0:" zur Adresse 9 geschickt.

KEINE DER BEKANNTEN NACHTEILE DURCH DOPPELFLOPPER

Dies hat auch wieder zwei Nachteile: Erstens muß es jedesmal neu geladen werden und zweitens ist solch eine Lösung mit den wenigsten Programmen (z.B. Wordpro) kompatibel. Ein Anschluß zweier Floppies als Adresse 8 ist nicht möglich, da hier dann beide gleichzeitig versuchen, Daten zu senden und ein heilloses Durcheinander entsteht.
All die Nachteile vermeidet Ihnen Doppelflopper 64, eine kleine Hardware-Lösung, die Ihnen auf Tastendruck jeweils ein Laufwerk als Adresse 8 selektiert, welches Sie dann bei der folgenden Schreib- oder Leseoperation ansprechen. Und der Doppelflopper ist garantiert 100% kompatibel zu sämtlichen Programmen und Hardware-Erweiterungen zum C 64, auch Floppy-Speedern und dergleichen!
Zur Funktion des Doppelfloppers: Wie Sie sicher wissen, gibt es in der 1541 zwei Brücken, die die Geräteadresse festlegen. Der Doppelflopper beruht auf der Idee, an einer dieser Brücken einen Schalter anzubringen, der zwischen Adresse 8 und 9 umschaltet. Nun hat die Sache einen Haken: Solch ein Schalter ist an sich wirkungslos, da die Brükken nur beim Setup der Floppy (z.B. Einschalten) überprüft werden und die Geräteadresse daraufhin entsprechend eingestellt wird. Nun ist ein Aus- und gleich wieder Einschalten der Floppys nicht immer gut für inliegende Disketten, ganz abgesehen von Hardware-Schäden, die durch häufiges Schalten auftreten können. Einen Schritt eleganter gehen wir vor, wenn wir einen RESET auslösen, der ja ein Setup der Floppy bewirkt und eben auch die Geräteadresse neu einstellt. Einen kleinen Nachteil hat die Sache noch: Beim RESET passiert fast schon zuviel – die Floppy läuft kurz an, die LED leuchtet und das alles dauert ca. drei Sekunden.
Betrachtet man sich das Betriebssystem der Floppy einmal genauer, so entdeckt man noch einen Hardware-Interrupt, der die Geräteadresse überprüft: den NMI. In diversen Handbüchern ist zwar zu lesen, er werde vom System nicht benutzt – das hindert uns aber nicht daran, es zu tun! Der NMI springt über $FFFA, $FF01 und $ FEE7 zu $EB22 und vollbringt eine Art "Mini-Setupt", der vielleicht eine Hundertstel-Sekunde benötigt und sich gar nicht bemerkbar macht.
Nun bauen wir uns einen Schalter, besser einen Taster (siehe Abb. 1), der bei den Floppies gleichzeitig einen NMI auslöst, der dann abhängig davon, welche Taste wir gedrückt haben, eine Floppy als Gerät 8 deklariert und die jeweils andere als 9. Ein NMI wird (ähnlich dem RESET) durch Legen des NMI-Pins gegen Masse aktiviert. DEV (und Masse) bilden die Brücke, welche über Adresse 8 oder 9 entscheidet. Drücken Sie nun einen der beiden Taster, wird diese Brücke bei einer Floppy geschlossen und gleichzeitig ein NMI ausgelöst. Sofort wird diese Floppy aktiviert (Adresse 8) und die andere auf Wartestellung geschaltet (Adresse 9).
Für jeden, der mit einem Lötkolben umgehen kann, dürfte der Aufbau der Schaltung (Abb. 2) keine



Zum komfortablen Diskettenhandling sind zwei Laufwerke unentbehrlich. Mit unserer Bastelei können Sie dies nun auch unter der selben Geräteadresse tun.

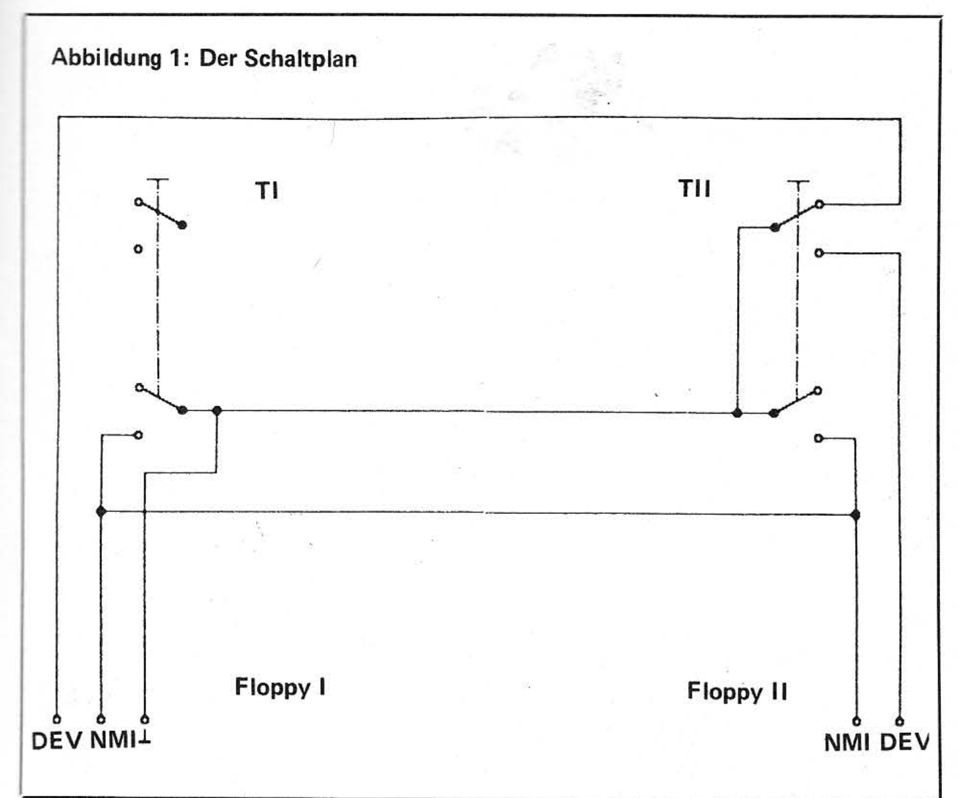


Abbildung 1: Der Schaltplan

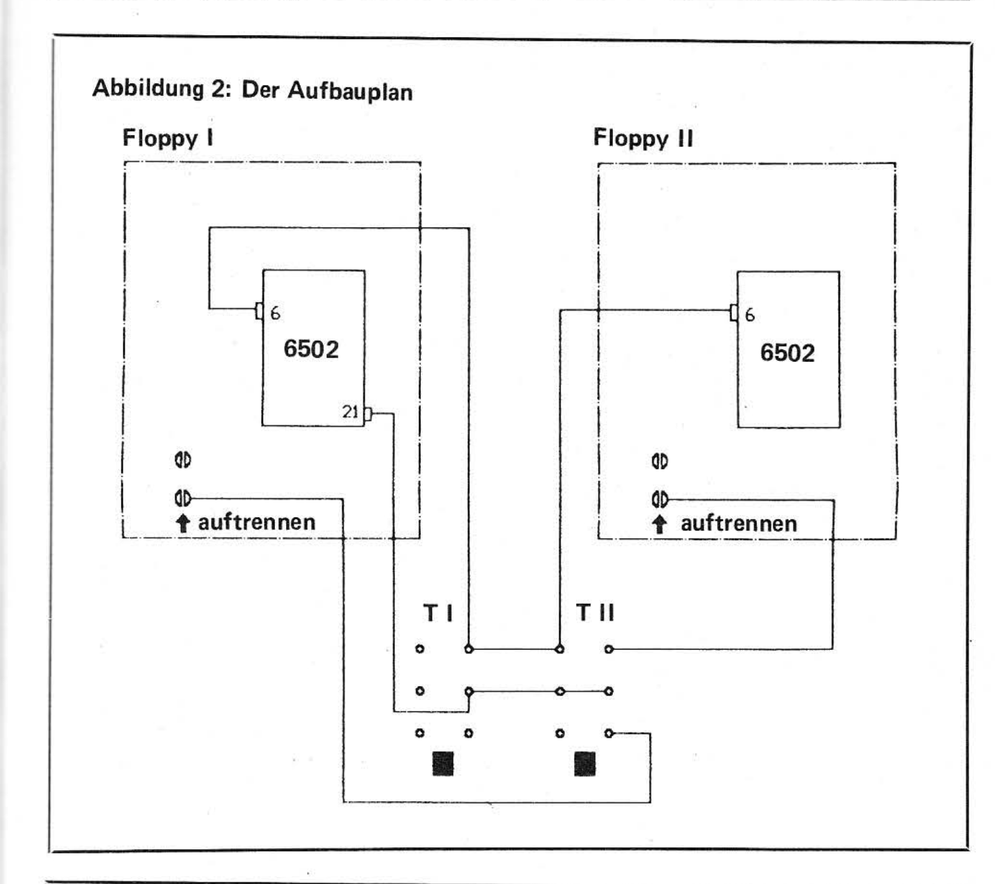


Abbildung 2: Der Aufbauplan

Stückliste:

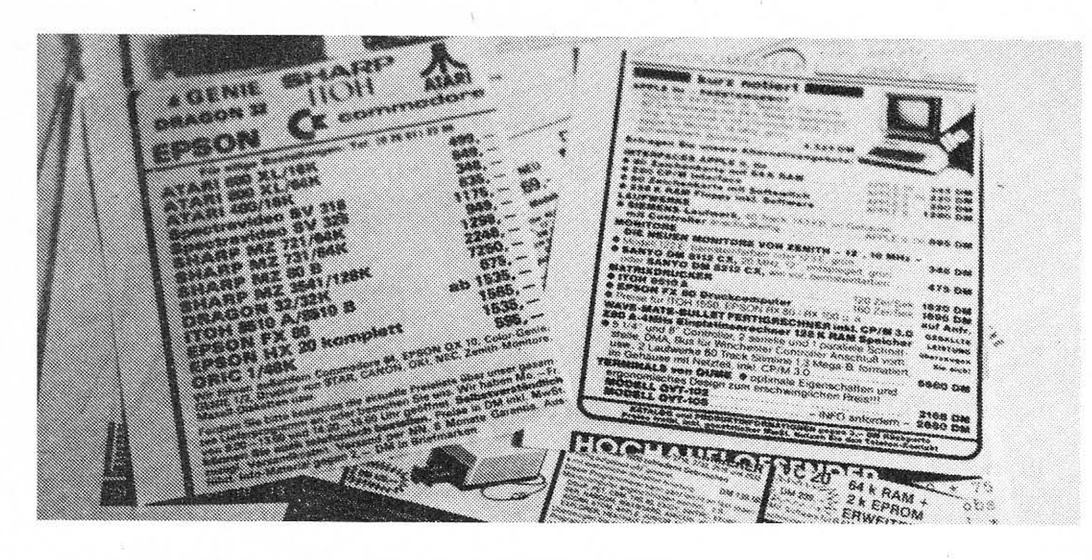
2 Drucktaster 2x µm
1 kleines Gehäuse
5 dünne, isolierte Drähte, ca. 50 cm
3 isolierte Drähte, ca. 3 cm
etwas Lötzinn
Materialkosten: ca. DM 5,–

Probleme bereiten. Sie schalten beide Floppies ab, stecken alle Kabel aus und heben die Deckel ab, nachdem Sie die je vier Schrauben am Boden der Gehäuse entfernt haben. Stellen Sie die Deckel senkrecht neben die Laufwerke. Nun durchtrennen Sie die beiden vorderen, mit Pfeil gekennzeichneten Brükken mit einem kleinen Schraubenzieher oder Dorn (mit Ohm-Meter überprüfen!). Löten Sie nun jeweils an Pin 6 der CPU 6502 (NMI) und am rechten Teil der vorderen Brükke (DEV) einen Draht an. Dabei gehen Sie folgendermaßen vor: Zuerst benetzen Sie die Lötstelle mit einem Tropfen Lötzinn, dann verzinnen Sie die abisolierten Kabelenden, um sie dann sogleich anzulöten. Zusätzlich kommt an Pin 21 der ersten Floppy (Masse) ein Draht. Achten Sie darauf, den Chip nicht zu überhitzen! Die Kabel führen Sie über dem Netzstecker oder durch die Lüftungsschlitze im Gehäuse-Deckel nach außen. Die Taster fin-

VORSICHTIG LÖTEN IST WICHTIG!

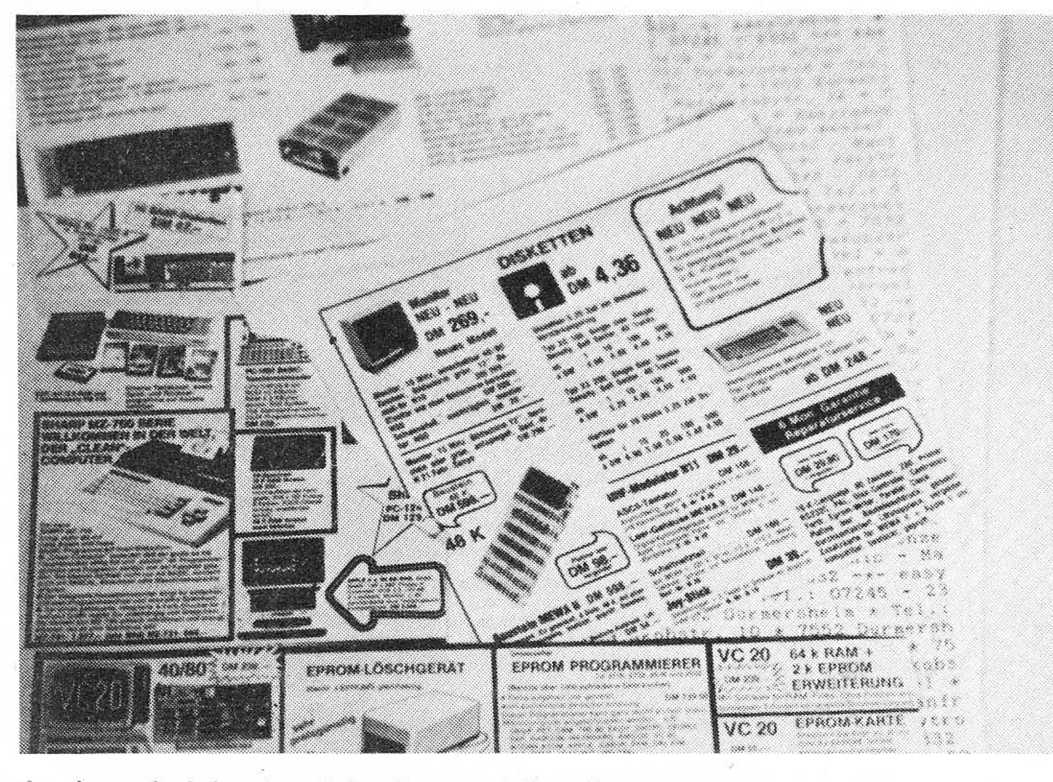
den in einem kleinen Gehäuse Platz (oder über einem Winkel direkt am Regal, wie bei mir). Löten Sie die Drähte genau wie in Abb. 2 gezeigt, an den Tastern an und vergessen Sie auch nicht die drei Verbindungen der Schalter untereinander. Ist alles erledigt, setzen Sie die Deckel vorsichtig wieder auf (klemmen Sie kein Kabel ein) und bringen die Schrauben wieder an. Kabel dran, einschalten und los geht's. (Unmittelbar nach dem Einschalten hat Floppy 1 die Adresse 8, was sich durch Drücken von T II sogleich ändert . . .)

Noch einige kleine Warnungen zum Abschluß: Versuchen Sie nie, während eines Schreib- oder Lesevorgangs einen Taster zu betätigen! Es würde zwar nichts beschädigt, ein Absturz des Systems ist Ihnen aber sicher. Beachten Sie desweiteren, daß selbst dieser kleine Eingriff die Garantie zum Erlöschen bringt! Und – wagen Sie sich nur ans Bauen, wenn Sie etwas Erfahrung im Löten haben. Lassen Sie es notfalls einen Freund erledigen. Und wenn Sie keinen solchen Freund haben: Der Autor dieser Schaltung erklärt sich gegen eine Gebühr von DM 75 bereit, Ihnen einen Doppelflopper einzubauen, wobei allerdings das Transportrisiko von Ihnen zu tragen wäre.



COMPUTER-VERSANDHANDEL

Die Angebote des Versandhandels sind preislich oft verlockend. Warum ist diese Handelssparte meist so preisgünstig und worauf sollten Sie beim Kauf im Versandhandel achten?



Anzeigen, wie sie in nahezu jedem Computerheft zu finden sind

Mittlerweile findet man sie in jeder Computerzeitschrift, teilweise sogar in technischen Magazinen, die mit Computer selbst eigentlich gar nichts zu tun haben.
Die Rede ist von den Anzeigen des Computerversandhandels; von Firmen, die sich darauf spezialisiert haben, ihre Produkte über den Postversandweg gegen Vorkasse oder per Nachnahme zu vertreiben. Oftmals haben sie noch nicht einmal ein Ladenlokal; sie bestehen nur aus einem Lager, einem Telefon und einem Briefkasten.
Dennoch sind sie für andere Computerhändler eine äußerst ernst zu nehmende Konkurrenz.
Der Grund hierfür liegt in den zum Teil enorm günstigen Preisen.
Manche Versandbetriebe unterbieten den empfohlenen Verkaufspreis um bis zu 30 (dreißig) Prozent!!!
Bei einem Drucker, der normalerweise 1800 DM kosten würde, macht das einen Preisunterschied von 600 DM oder mit anderen Worten einen Endpreis von nur noch 1200 DM aus!
Wen wundert es da, wenn so mancher kleine Computerladen ca. 2 – 3 Jahre nach seiner Eröffnung wieder schließt, wird doch im Versandhandel ein Preiskampf betrieben, der schlichtweg als ruinös zu bezeichnen ist.
Unlängst ist es sogar vorgekommen, daß der Einkaufspreis eines Händlers direkt vom Werk ab höher lag, als der Endpreis für den „normalen" Endverbraucher selbst!
Wie aber kommt es nun dazu, daß gerade im Versandhandel die Preise derart herabgesetzt sind?
Als ersten Grund für diese Niedrigpreise kann man hier die gegenüber dem Computerfachhandel schlechte Beratung anführen. Geht man in einen Com-

puterladen, so kann man sich dort beraten lassen, die dort vorhandenen Vorführgeräte testen; man hat sie eben vor sich stehen.
Im Gegensatz zum Versand sieht man das Produkt, das man kaufen will, direkt vor sich, man hat den Verkäufer gleich in nächster Nähe zur Hand, und kann ihm, bei etwaigen Defekten am Gerät während der Garantiezeit, gewaltig „auf den Schlips treten". (Was leider oft genug nötig ist!!!)
Daß in solchen Läden der Preis gegenüber dem des im Versandhandels höher sein muß ist logisch, bedenkt man, welche Kapitalwerte in Form von Vorführgeräten in einen Computerladen zinslos gesteckt werden müssen.
Überdies muß in einem Laden immer mindestens eine Bedienung sein, woraus folgt, daß mindestens ein Angestellter zu bezahlen ist, egal wie das Geschäft läuft – und das kann teuer sein.
Im Gegensatz hierzu benötigt der Versandhandel keine Vorführgeräte, keine Vorführräume, kein extra geschultes (und teures) Vorführpersonal.
Sämtliche Beratungsstunden entfallen, mit Ausnahme von kurzen Tips am Telefon, gänzlich; der Zeitaufwand für den Verkauf ist alles in allem daher auch entsprechend gering.
Doch dies ist nicht der einzige Vorteil des Versandhandels gegenüber dem „normalen" Fachhandel.
So hat eine Computerfirma, die in einer Computerfachzeitschrift wirbt, die gesamte BRD, Österreich, Luxemburg und den deutschsprachigen Teil der Schweiz abgedeckt; das heißt: Die Kundschaft ist praktisch der gesamte deutschsprachige Raum in Europa.
Die Versandfirmen sind nicht so sehr wie die Computerfachhandelsniederlassungen von der Konjunktur an ihrem Ort abhängig. Ein Computerladen in Hamburg hat zwar in den Bewohnern des gesamten Hamburger Raumes und des Hamburger Umfeldes potentielle Kunden, doch kann er sich damit nicht mit der Zahl des Versandhandels messen.
Hieraus wächst der zweite große Vorteil:
Durch die größere Anzahl an Kunden werden mehr Artikel verkauft.
Mehr verkaufte Artikel bedeuten einen größeren Gewinn, so daß man die Preise senken kann, um noch mehr Produkte zu verkaufen.

So mancher Laden schließt schnell

Verkauft man nun tatsächlich noch mehr Produkte, so erhält man durch die höheren Abnahmemengen bei den Herstellerfirmen größere Einkaufsnachlässe und kann somit wieder größere Gewinne einstekken sowie eventuell weiterhin den Preis senken.
Diese Preissenkungspolitik, die dem Endverbraucher bisher anscheinend sehr zu Gute gekommen ist, führt zu immer größeren Verkaufsstückzahlen bei immer kleineren Gewinnen je verkaufte Einheit.
Bleiben dann einmal aus irgend einem Grund die gewohnten hohen Verkaufsstückzahlen aus, kann der Betrieb leicht Pleite gehen, denn dann fehlen ja die nötigen Gewinne, um entstandene Fixkosten zu decken.
Unsicherheit über die Lebensdauer eines Versandbetriebs ist ein genereller Minuspunkt gegenüber dem Ladenlokalhandel.
Zwar sprießen Ladenlokale im Bereich Computerverkauf wie die Pilze bei lauwarmem schwülem Wetter aus dem Boden und es werden viele auch schon bei dem ersten größeren Regen wieder weggeschwemmt, doch die Schließungsquote solcher Ladengeschäfte liegt im Vergleich mit der des Versandhandels ungleich niedriger, was eine höhere Sicherheit vor Konkurs beim Fachhandel aufzeigt.
Schließlich verliert man bei einem vollständigen Zusammenbruch einer Firma oftmals sein Recht auf Garantie-Reparatur im Falle eines Defektes, beziehungsweise besser gesagt: Meist ist es schwieriger und kostspieliger, auf der Reparatur eines Gerätes im Garantiefall zu bestehen, wenn die Firma, bei der das Gerät gekauft wurde, mittlerweile Pleite gegangen ist, als wenn man die Reparaturkosten gleich privat, also zu eigenen Lasten, trägt.
Allgemein scheint im Durchschnitt der Service des Versandhandels um einiges schlechter zu sein als der des Fachhandels.
Natürlich, es gibt auch Versandfirmen, die defekte Geräte während der Garantiezeit anstandslos umtauschen.

Versandladen: Oft kein Service

Ebenso gibt es Fachhandelsbetriebe, die für die Reparatur zum Beispiel eines einzelnen Diskettencontrollers über 4 Wochen benötigen!!!, obwohl der Fehler am Gerät nun wirklich in weniger als einer halben Stunde hätte gefunden werden können.
Doch das ist bei einem Laden vor Ort viel seltener, denn dort kann man seinem Begehren auch viel besser Luft machen, als es zum Beispiel durch einen Mahnbrief oder Telefonanruf an eine Versandfirma möglich ist.
Zusammenfassend kann ich betreffend des Thema Reparatur während der Garantiezeit sagen, daß der Versandhandel hierbei durchaus 2 – 3 Monate (1/4 Jahr!) benötigt, um ein defektes Teil zu reparieren oder auszutauschen.
Wer mir das nicht glaubt, kann bei mir Belege anfordern, die meine Behauptung selbstredend beweisen, denn ich habe mit einer dieser „Schlamper"-Versandfirmen meine eigenen Erfahrungen gemacht.

Beim Erstkauf besser zum Fachhandel

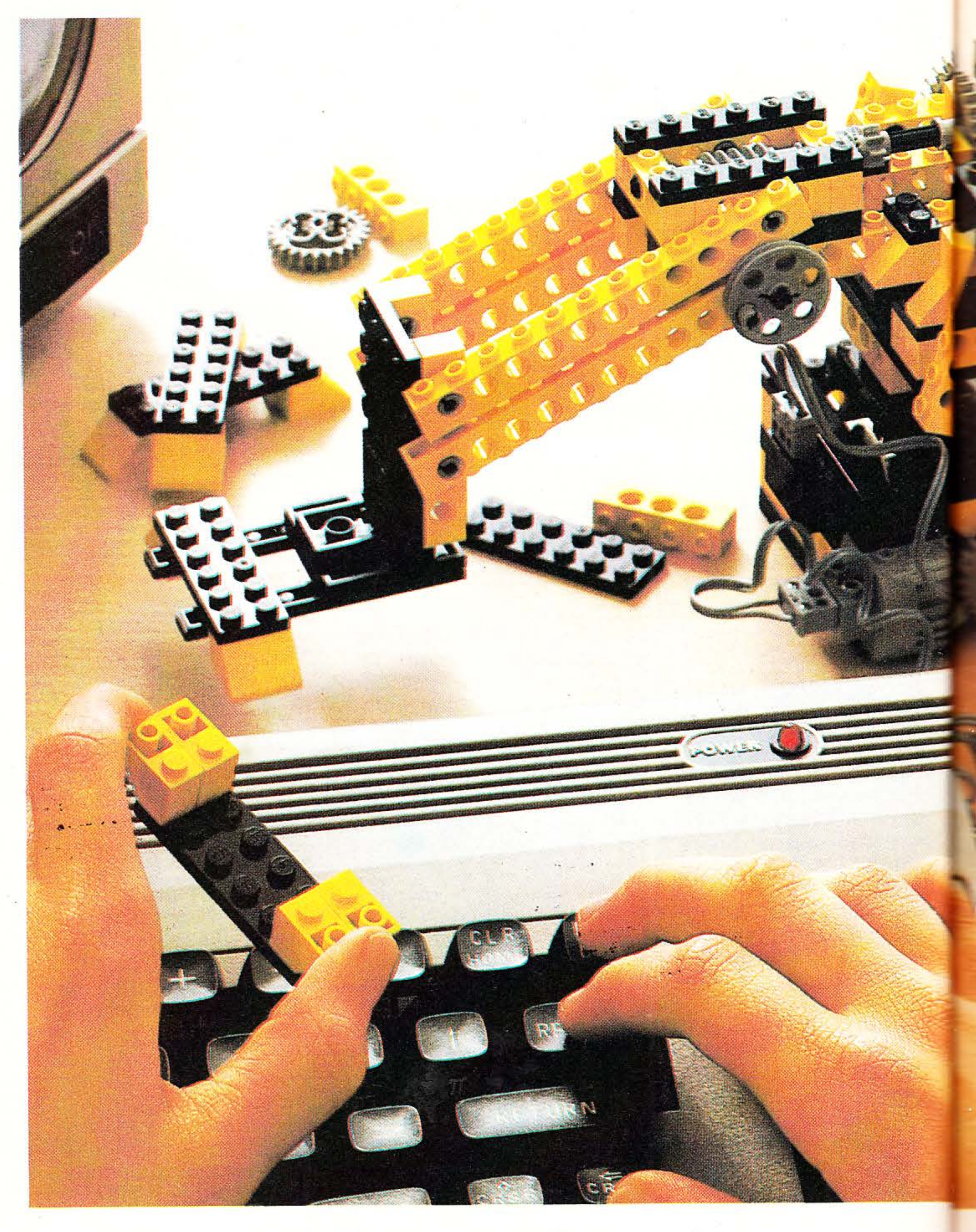
Daher rate ich im allgemeinen folgendes:
Wenn Sie Anfänger sind, also von der Computerei noch nicht allzu viel wissen, sollten Sie Ihre Grundgerätekonfiguration beim Fachhändler in nächster Nähe kaufen, denn dort sollten Sie zumindest beraten werden und Sie können hier auch mit deutlich kürzeren Wartezeiten rechnen, falls Ihr Gerät einmal defekt sein sollte.
Hat man sich allerdings schon mit EDV beschäftigt, und kennt man sich gut mit ihr aus, so kann eher empfohlen werden, das Gerät doch im Versandhandel zu kaufen.
Allerdings sollte der Preisunterschied schon größer als 10 Prozent sein.
Erweiterungen für das Gerät (z.B. Disketten) können ruhig über den Versand erworben werden.
Man erhält sie hier meist unschlagbar billig (wenn man das richtige Angebot gefunden hat) und die Ausfallquote liegt extrem niedrig.
Überdies riskiert man bei Erweiterungen bzw. Zubehör auch nicht allzu viel Geld, so daß man mit diesem Kauf auch eine Versandfirma z.B. auf Lieferzeiten hin testen kann.
Wir hoffen, daß wir mit diesen Tips dem einen oder anderen Leser geholfen haben, sich zu entscheiden und wünschen allen noch viel Spaß mit der EDV. *Elke Knoll*

IHR COMPUTER

Zwei Programme zusammenzufügen, scheint auf den ersten Blick unmöglich. Doch auch hier gibt es Abhilfe, wie der folgende Bericht beweist. Außerdem erläutern wir wichtige Adressen des Tedmon im C 16/P 4.

Wenn Sie in Ihren Computer Programmzeilen eingeben oder im Direktmodus arbeiten, dann werden diese zwischengespeichert. Ihr Computer besitzt sowohl einen Eingabepuffer als auch einen Tastaturpuffer. Der Eingabepuffer liegt im Bereich von $200 (dezimal 512) bis $258 (600) und der Tastaturpuffer von $527 (1319) bis $530 (1328). Mit dem TEDMON können Sie sich von der Existenz des Eingabepuffers überzeugen. Dabei ist der Tastaturpuffer weitaus interessanter für uns, wie Sie gleich feststellen werden. Dieser Puffer in Verbindung mit einer weiteren Speicherstelle ermöglicht es, dem Computer vorzugaukeln, wir hätten verschiedene Tasten gedrückt. Was das soll? Nun, dadurch kann der Computer trickreich überlistet werden. Stellen Sie sich doch einmal vor, daß ein Programm Tastendrücke simuliert und dann mit dem Programmlauf abbricht. Stellen Sie sich weiterhin vor, als "Tastendrücker" wäre die Zeichenfolge RUN (RETURN) in den Tastaturpuffer geschrieben worden. Wie würde sich der Computer verhalten? Falsch, er würde nicht wieder starten. Sie haben es bestimmt so erwartet, denn wenn wir so etwas ansprechen, dann steckt doch etwas dahinter. Es ist auch so, der Computer kann wirklich soweit gebracht werden, das, was in den Tastaturpuffer geschrieben wurde, auszuführen. Es muß ihm nur mitgeteilt werden, wieviele Tasten gedrückt wurden und diese Information muß in die Speicherstelle für die Anzahl der gedrückten Tasten gePOKEt werden, dann klappt's. Die hierfür zuständige Speicherstelle ist an der Adresse $EF (239) zu finden.

"Grau ist alle Theorie", deswegen gleich zur Praxis. Bild 1 zeigt ein kleines Programm, welches in ganz einfacher Weise aufzeigt, wie es geht. Durch Zeile 140 wird der Bildschirm gelöscht. Durch 150 bis 160 wird nach dieser



Bildschirmbereinigung das Wort MONITOR ausgegeben. Dieses Wort ist, wie Sie wissen, der Befehl zum Aufruf des TEDMON. Zeile 170 setzt den Cursor auf die Home-Position, also in die linke obere Ecke des Bildschirms, ohne ihn zu löschen. Und nun folgen die beiden Programmzeilen, die den Computer überlisten. POKE 239,1 bewirkt, daß der Computer nach seinem Programmlauf glaubt, eine Taste wurde gedrückt und POKE 1319,13 schreibt den Code für die Return-Taste in den Tastaturpuffer. Nach dem Programmlauf meldet sich der Computer mit READY und sein Cursor steht auf dem Wort MONITOR. Durch seinen Full-Screen-Editor wird dieses Wort nun "übernommen" und der Befehl ausgeführt. Das war nun die einfachste Version: Ein Tastendruck wurde simuliert. Bild 2 zeigt dann eine bereits erweiterte Variante hiervon. In den Tastaturpuffer wird zweimal der Wert der Return-Taste geschrieben und die Anzahl der gedrückten Tasten ist ebenfalls zwei (POKE 239,2).



Mittels der bisher aufgezeigten Methode können sehr lange Befehlsfolgen vom Bildschirm übernommen werden. Es gibt aber auch die Möglichkeit, ohne den Bildschirmspeicher zu arbeiten, indem die Zeichencode der "Eingabe" in den Tastaturpuffer geschrieben werden. Bild 3 zeigt, wie es geht. In den Datastatements der Zeile 330 stehen die Werte für das Wort MONITOR und der Code der Return-Taste. Aufgrund unserer bisherigen Ausführungen dürfte eigentlich kein Verständnisproblem bei Ihnen auftreten. Chr$ (77)=M, Chr$ (79)=0 usw. Auch diese Methode eignet sich dazu, direkt eine weitere Eingabe zu simulieren, also so zu tun, als wenn Sie nicht nur MONITOR eingegeben hätten, sondern auch noch "M" (RETURN). Bild 4 zeigt nochmals das komplette Programm dafür.

Bisher haben wir nur gezeigt, wie diese Technik funktioniert. Nun wollen wir an einem kleinen Beispiel aufzeigen, wie dies sinnvoll eingesetzt werden kann. Auch hierzu wieder ein kleines Programm (Bild 5). Dieses

DIESE METHODE HAT SINN

Programm soll demonstrieren, wie beispielsweise neue Basiczeilen erzeugt werden können. Diese Methode wäre beispielsweise für eine simple kleine Adressdatei einsetzbar. Die gleiche Methode kann auch dafür eingesetzt werden, einen Datagenerator zu programmieren, um Maschinenspracheprogramme als Hexlader abdrukken zu können. In einer unserer regulären Commodore-Welt ist ein derartiges Programm für den C16 bis Plus4 enthalten!

WIR FÜGEN PROGRAMME ZUSAMMEN

Doch nun zum versprochenen Thema der Programmzusammenfügung. Warum versuchen wir immer wieder, Ihnen die hexadezimale Denkweise nahezulegen? Vielleicht haben Sie es schon bemerkt, mittels des eingebauten Monitors können nicht nur Programmanalysen durchgeführt werden, sondern auch sehr sinnvolle Manipulationen sind schnell durchführbar. Vorausgesetzt, man kann mit Hexzahlen arbeiten.

Nehmen wir also einmal folgenden Fall an: Sie möchten zwei Programme miteinander zu einem einzigen verbinden. Handelt es sich bei einem von beiden um ein sehr kurzes Programm, so gibt es (wenn beide Programme unterschiedliche Zeilennummern haben) eine sehr einfache und simple Möglichkeit. Lassen Sie sich das kleine (nur aus ein paar Zeilen bestehende) Programm durch LIST auf dem Bildschirm ausgeben. Danach laden Sie das zweite Programm ganz normal. Wenn nach dem Ladevorgang das erste Programm noch auf dem Bildschirm steht, dann können Sie dieses nun ganz einfach dadurch übernehmen, daß Sie mit dem Cursor auf die erste Basiczeile des ersten Programmes fahren und dann alle diese Zeilen durch Betätigung der RETURN-Taste übernehmen. Sie können dabei auch – falls es erforderlich ist – die Zeilennummern ändern.

Der Bildschirmspeicher fungiert also als Zwischenspeicher. Falls aber keines der beiden Programme so kurz ist, daß es (nach dem Ladevorgang des zweiten Programmes) noch komplett auf dem Bildschirm steht, dann muß man eben eines der Programme in kleine Teile zerlegen und den Vorgang mehrfach wiederholen. Eine ziemlich umständliche Vorgehensweise, finden Sie nicht auch?

Leider besitzen die Commodore - Computer der Serie C116/16 und Plus4 keinen MERGE-Befehl, mit dem es möglich wäre, zwei Programme miteinander zu verketten. Aber das ist kein allzugroßes Manko, denn nun erfahren Sie, wie man dieses Problem innerhalb sehr kurzer Zeit lösen kann. TEDMON hilft dabei!

Immer wieder haben wir Ihnen Speicherstellen mitgeteilt, in denen sich Ihr Computer etwas merkt. Für die Manipulationen, die wir nun vor- »

haben, sind genau vier davon interessant. Diese befinden sich ab $2B (dezimal 43) bis $2E (45) hintereinander im Speicher.

In den ersten beiden Speicherstellen steht die Adresse des Basic-Programmbeginnes (Zeiger auf den Basic-Anfang), in den beiden folgenden der Zeiger für den Beginn der Variablen. Erinnern Sie sich noch an das Programm STRINGFINDER in der letzten Ausgabe? Bei diesem Programm erfuhren wir die Anfangs- und Endadressen für den Programmlauf.

Nun aber "gebrauchen" wir sie wieder anders, wir wollen eigentlich gar nicht wissen, wo Basic-Beginn und Start der Variablen ist, sondern einen dieser Zeiger verstellen.

Wenn Sie Ihren Computer einschalten und dann einmal nachsehen (ohne ein Programm geladen zu haben), was in den Speicherstellen ab $2B steht, dann finden Sie folgende Zeichenfolge: 01 10 03 10.
Falls Sie noch nicht wissen sollten (es soll ja Besitzer dieser Computer geben, die bisher unsere Zeitschriften noch nicht gelesen haben!), wie Sie sich diese Speicherstellen ansehen können, hier der ganz kurze Hinweis: Ihr Computer hat einen eingebauten Maschinensprache-Monitor, der durch den Befehl MONITOR aufgerufen werden kann. Nach diesem Aufruf geben Sie "M2B" ein und schon sehen Sie ab dieser Adresse die Speicherinhalte.

Die ersten beiden Bytes geben den Basic-Beginn an. "Basic" und damit geladene Basicprogramme beginnt bzw. beginnen also bei der RAM-Speicher-Adresse $1001. Die erste Zahl ist das sogenannte Low-Byte, die zweite das High-Byte. Um den hexadezimalen Wert, auf den der Zeiger "deutet", zu erhalten, muß die Zeichenfolge einfach umgedreht werden. Für den Zeiger auf den Beginn der Variablen ergibt sich deshalb die Adresse $1003.

Warum dieser Abstand vorhanden ist, hängt mit der Programm-Ende-Erkennung des Basic-Interpreters zusammen. Prinzipiell ist die Differenz zwischen dem eigentlichen Programmende und dem Start der Variablen immer 2 (Das Ende eines Basic-Programmes wird durch zwei leere Bytes gekennzeichnet, wir hatten das schon in der letzten Ausgabe mitgeteilt!). Da der Variablenbeginn immer direkt hinter dem Programm-Ende liegt, können wir durch einen kleinen Trick nun die Adresse auf den Basic-Beginn hinter das Programm legen.

PEEKEN UND POKEN IST VIEL ZU UMSTÄNDLICH

Dies erfolgt einfacherweise mit dem eingebauten TEDMONITOR. Um diese Manipulation korrekt durchzuführen, ziehen Sie von der Adresse des Zeigers auf die Variablen zwei ab und schreiben diesen Wert in die Speicherstellen für den Basicbeginn. Fertig! Damit Sie das aber korrekt tun können, müssen Sie mit den Hexzahlen umgehen können. Selbstverständlich geht es auch durch PEEKen und POKEn in Dezimal, ist aber weit umständlicher.

Nun können Sie das nächste Programm einladen und dieses wird hinter das erste einge-

```
100 REM AUFRUF DES MONITORS UEBER
110 REM BILDSCHIRMSPEICHER UND
120 REM TASTATURBUFFER (VERSION 1)
130 :
131 REM BILDSCHIRM LOESCHEN
140 PRINT CHR$( 147)
145 :
150 PRINT:PRINT
160 PRINT"MONITOR"
170 PRINT CHR$( 19)
180 POKE 239,1
190 POKE 1319,13

READY.
```

Bild 1

```
100 REM AUFRUF DER MONITORFUNKTION
110 REM 'M 1000' UEBER DEN BILDSCHIRM-
120 REM SPEICHER UND TASTATURPUFFER
130 REM (VERSION 2)
140 :
150 REM BILDSCHIRM LOESCHEN
160 PRINT CHR$( 147)
170 :
180 PRINT:PRINT:PRINT"MONITOR"
190 FOR I = 1 TO 5:PRINT:NEXT
200 PRINT"M 1000"
210 PRINT CHR$( 19)
220 POKE 239,2
230 POKE 1319,13:POKE 1320,13

READY.
```

Bild 2

```
100 REM AUFRUF DES MONITORS
110 REM DURCH MANIPULATION DES
120 REM TASTATURPUFFERS (VERSION 1)
130 :
140 REM BILDSCHIRM LOESCHEN
150 PRINT CHR$( 147)
160 :
170 REM TEXTAUSGABE
180 PRINT"AUFRUF DES MONITORS DURCH
190 PRINT"MANIPULATION DES TASTATUR-
200 PRINT"PUFFERS (VERSION 1)
210 :
220 :
230 REM DAS WORT MONITOR IN DEN
240 REM TASTATURPUFFER SCHREIBEN
250 FOR I = 1319 TO 1326
260 READ A:POKE I,A
270 NEXT
280 :
290 REM ANZAHL SIMULIERTER TASTENDRUECKE
300 POKE 239,8
310 :
320 REM   M  O  N  I  T  O  R
330 DATA 77,79,78,73,84,79,82,13

READY.
```

Bild 3

lesen. Nachdem das zweite Programm geladen wurde, muß nun wieder der Zeiger für den Basicbeginn korrigiert werden und schon kann das Gesamtprogramm aufgelistet werden. Der Basicbeginn ist im Normalfall ja immer $1001 und deshalb kann ohne weitere mathematische Klimmzüge dieser Wert sofort eingeschrieben werden.

Es kommt nun aber vor, daß die beiden zusammengefügten Programmteile gleiche Zeilennummern haben. Wenn Sie sich in einem solchen Falle das komplette Programm-Listing ausgeben lassen, dann sehen Sie beispielsweise die Basiczeilen von 100 bis 500 des ersten Programmes und dahinter folgend dann (vielleicht bei 10 beginnend) die Zeilen des zweiten Programmes. Das ist aber kein Beinbruch. Der Befehl RENUMBER schafft wieder Ordnung und korrigiert alles. Damit haben wir eigentlich erklärt, was alles zu machen ist. Damit es aber bei Ihnen keine Probleme gibt, spielen wir ein praktisches Beispiel durch.

Angenommen, folgende beiden Programme sollen zu einem zusammengefügt werden:

PROGRAMM 1

```
10 REM DIES IST DER ERSTE TEIL
20 REM DES SPÄTEREN GESAMTPROGRAMMES.
```

PROGRAMM 2

```
10 REM UND DIES IST DER ZWEITE TEIL.
15 REM EINZELPROGRAMME ZU EINEM
20 REM GESAMTPROGRAMM ZUSAMMENZUHÄNGEN
30 REM IST KEIN PROBLEM!
40 REM WICHTIG IST; GEWUSST WIE!
```

Zunächst geben Sie das erste Programm in Ihren Commodore ein und speichern es ab. Beispiel für Besitzer einer Diskettenstation: DSAVE"P1"

Danach geben Sie das zweite Programm ein und speichern es ebenfalls ab. Beispiel DSAVE"P2". Falls keine Diskettenstation vorhanden ist, dann hat der Abspeichervorgang ganz normal mit SAVE "Name des Programmes" zu erfolgen.

Damit Sie nun ganz sicher sind, daß die Ihnen nun gezeigte Technik keinen verspäteten Aprilscherz darstellt, schalten Sie den Commodore nun aus und wieder ein. Programme, die im Speicher standen, sind dadurch mit Sicherheit gelöscht.

Nun laden Sie das erste Programm. Wenn der Ladevorgang beendet ist, rufen Sie den Monitor auf. Also: MONITOR eingeben und die Return-Taste drücken.

Danach "M2B" eingeben und wieder die Return-Taste drücken. In der Zeile, in der links die Adresse > 002B steht, sehen sie nun die Zeichenfolge: 01 10 47 10 usw. Fahren Sie nun mit dem Cursor auf die "01" und geben Sie dort die Zahl 45 ein. (Hexadezimal 47 minus 2 = 45!) Nach Betätigung der Returntaste wird diese Eingabe übernommen.

Nun können Sie mit X (Return) den Monitor verlassen. Wie Sie sich durch LIST überzeugen können, wird kein Programm mehr ausgegeben. Nun laden Sie das zweite Programm. Nach dem Ladevorgang den Monitor wieder aufrufen und M2B eingeben. Durch Eingabe von 01 an der Adresse S2B, @ wird der Basicbeginn wieder auf den ursprünglichen Wert gestellt. Wenn Sie dann den Monitor verlassen, können Sie das Gesamtprogramm auflisten. Wie Sie sehen, wurden beide Programme zu einem einzigen zusammengehängt, aber sehen Sie sich bitte auch einmal die Zeilennummern an! Deshalb bitte noch neu durchnumerieren und alles stimmt wieder.

Nun steht Programmzusammenfügungen nichts mehr im Weg. Sie sehen, wenn man seinen Computer etwas kennt, sind Dinge machbar, die er sonst nicht beherrscht. Im nächsten Heft geht es wieder weiter mit Informationen, damit Sie Ihren Computer noch besser kennenlernen und damit auch trickreicher programmieren. *(LM)*

IHR COMPUTER KANN MEHR

```
100 REM AUFRUF DES MONITORS
110 REM UND START EINES HEXDUMPS
120 REM DURCH MANIPULATION DES
130 REM TASTATURPUFFERS (VERSION 2)
140 :
145 REM BILDSCHIRM LOESCHEN
150 PRINT CHR$( 147)
151 PRINT"AUFRUF DER MONITORFUNKTION 'M'
153 PRINT"DURCH TASTATURPUFFER-
154 PRINT"MANIPULATION (V 2)
160 :
170 REM DAS WORT MONITOR, (RETURN)
180 REM UND DEN BUCHSTABEN 'M' IN DEN
190 REM TASTATURPUFFER POKEN
200 FOR I = 1319 TO 1328
210 READ A:POKE I,A
220 NEXT
230 :
240 REM ANZAHL DER SIMULIERTEN TASTEN
250 POKE 239,10
260 :
270 REM   M  O  N  I  T  O  R  RETURN
280 DATA 77,79,78,73,84,79,82,13
290 REM   M  RETURN
300 DATA 77,13

READY.
```

Bild 4

```
100 REM DEMONSTRATIONSPROGRAMM ZUR
110 REM MANIPULATION DES TASTATURPUFFERS
120 :
130 PRINT CHR$( 147):A=1000
140 HO$=CHR$( 19):CD$=CHR$( 17):SCNCLR
150 PRINT"EINGABE 'ENDE' BEENDET ";
160 PRINT"DEN PROGRAMMLAUF":PRINT
170 INPUT"EINGABE";A$:PRINTCHR$( 147)
180 IF A$="ENDE" THEN END
190 PRINT HO$CD$CD$CD$A" DATA "A$
200 PRINT"A="A;"+10":PRINT CD$
210 PRINT"GOTO 140":PRINT HO$:POKE 239,3
220 FOR I=1319 TO 1321:POKE I,13:NEXT I

READY.
```

Bild 5

SPRACHBARRIEREN

Ein Commodore-Computer mit dem Basic 2.0 versteht sehr selten ein Programm in Basic 3.5 oder 7.0. Ein Commodore-Computer mit Basic 3.5, wie der C16, versteht sehr selten ein Programm für den C64, da dieser ihm fehlende Basicbefehle wie z.B. "joy" oder "color" mit PEEK, POKE- oder Sysbefehlen umgeht, wobei sehr selten die entsprechenden Adressen oder die dort einzutragenden Werte übereinstimmen. Ein Basic-Programm von einem Computer einer anderen Marke auf einen Commodore-Computer zu übertragen, ist zwar nicht unmöglich, setzt jedoch umfangreiche Systemkenntnisse voraus und ist mit erheblichem Arbeitsaufwand verbunden. Jedoch gibt es einen Weg, wie sich solche Arbeiten erübrigen. Die Lösung heißt: Macro-Prozessor.

Bereits Mitte der achtziger Jahre, als viele Computersprachen sich als ziemlich schwerfällig im Umgang mit komplexeren Anwendungen erwiesen, erschien als Lösung das Konzept des Hochsprachen-Macroprozessors. Die Verwendung von Macros war beileibe kein neues Konzept. Macroassembler gab es schon seit etlichen Jahren. Dann aber wurden Macros auch in Hochsprachen eingesetzt, um die Qualität der Programme zu verbessern und deren Entwicklungsdauer zu vermindern. An einem

DEFINITION EIGENER ROUTINEN

Beispiel läßt sich so etwas leicht deutlich machen. Nehmen wir an, Sie hätten eine Routine entwikkelt, die einen String "X$" zentriert ausdruckt und würden dieselbe mit einem "gosub" aufrufen. Wenn Sie mehrere Subroutinen verwenden, so wissen Sie bei mehrmaligem Renumbern garantiert nicht mehr so ohne Weiteres, welche Routine nun welche Zeilennummer hat. Dies wirkt sich auf die Lesbarkeit des Programmes und auf den weiteren Durchblick äußerst ungünstig aus. Hätten Sie dagegen die Möglichkeit, sich beispielsweise zu definieren:
!def:printz(x$)=gosub 320
und es würde fortan im Programm anstatt einer längeren Routine oder eines Sprunges auf irgendeine Zeilennummer nur ganz einfach printz(x$) oder auch printz(y$) zu lesen sein, wieviel an Übersichtlichkeit wäre dadurch doch schon gewonnen.
Ein anderes Beispiel: Ihr Freund hat keinen Commodore-Computer, sondern einen anderen. Bei ihm läßt sich der Cursor nicht mit:
char,z,s," " sondern mit gotoxy (s,z) positionieren. Sie einigen sich mit ihm nun ganz einfach, daß es bei Ihnen beiden nun statt dessen at(z,s) heißen soll.

UMWANDLUNG VON QUELLFILE IN PROGRAMMFILE

Wie ist es aber möglich zu erreichen, daß der Computer die Neudefinierung und ebenso die neuen Befehle versteht? Im Prinzip gibt es hierzu zwei Möglichkeiten. Die eine ist, das Betriebssystem Ihres Computers, in der Hauptsache ist hier der Basicinterpreter gemeint, dahingehend abzuändern. Das ist aber gewiß nicht jedermanns Sache, denn es setzt solide Maschinensprache und Systemkenntnisse voraus. Die andere ist, das Quellprogramm, welches die für den Computer unverständlichen Statements enthält, einem Übersetzungsprozeß zu unterziehen, dessen Ergebnis ein für den Computer aufbereitetes Programm darstellt, welches er versteht, welches auch wieder die ihm vertrauten Funktionen wie z.B. "char" beim C16 oder "gotoxy" beim anderen enthält, welches aber auch nicht mehr so schön zu lesen ist.
Das übersetzende Programm, ob es sich nun lediglich um einen Macroprozessor oder einen komfortableren Compiler handelt, liest den Quellfile und erzeugt mit Zugriff auf mehrere Bibliothekfiles, aus welchen er sich die definierten Routinen holt, einen ausführbaren Programmfile. Man braucht also nur mehr für die unterschiedlichen

MACRO-PROZESSOR-GRUNDLAGEN

Computer die unterschiedlichen Macroprozessoren und ebenso die unterschiedlichen Bibliothekfiles. Es können dann für die unterschiedlichen Systeme die gleichen Programme verwendet werden.
Am Anfang wollen wir es uns noch etwas einfach machen. Wir brauchen nicht gleich ein Superprogramm mit allen möglichen Extras. Besser, wenn wir mehr Wert auf das Verständnis der Grundlagen verwenden, damit Sie das Programm als Anregung aufnehmen und auch weiter ausbauen können.
Das Programm fragt als Erstes den Namen des zu bearbeitenden Files ab. Dieser, mit einem Zusatz versehen, ergibt den Namen des resultierenden ausführbaren Files.
Nach der Fileeröffnung kann es losgehen. Die Ladeadresse im Quellfile übergehen wir und schreiben, da wir einen C16/116/Plus4 unser eigen nennen, einfach $1001 in den Zielfile.
Die nächsten beiden Bytes die wir lesen, bilden die Linkadresse auf die nächste Basiczeile. Wir können diese getrost vergessen, da wir dieselbe sowieso neu berechnen müssen, weil sie nach dem ersten Macro nicht mehr stimmt.
Allerdings sehen wir uns vorher noch an, ob nicht etwa das Programmende erreicht ist, was man daran sieht, daß die Linkadresse Null ist. In letzterem Falle springen wir in die "Fileende-Routine". In der „Zeilennummer-Routine" lesen oder bestimmen wir die Zeilennummer. Dann lesen wir das erste Programmbyte in der aktuellen Basiczeile. Nun ist die Frage, was haben wir vor uns, eine ganz normale Basiczeile oder ein Macro? Dies läßt sich leicht klären, wenn wir die Übereinkunft treffen, daß ein Macro dadurch kenntlich sein soll, daß am Zeilenanfang gleich im Anschluß an die Zeilennummer ein "!" (Rufzeichen) stehen soll, es handelt sich ja auch um den Aufruf eines Macros. Liegt ein Macro vor, so wird dieses bearbeitet und dann mit der nächsten Basiczeile weitergefahren. Im anderen Falle lesen wir die aktuelle Basiczeile in einen String und speichern diesen mit richtig berechneter Linkadresse ab.

ZUGRIFF AUF MACROFILES

Nun kommt die große Frage, wie die Bearbeitung eines Macros überhaupt vor sich gehen soll, bzw. wie so ein Macro überhaupt aussehen soll. Im Prinzip sollten bei Macros Parameter übergeben werden können. Auch "GOTO" und "GOSUB"-Sprünge sollten möglich sein. Wir verzichten aber aus Gründen der Vereinfachung in diesem Artikel darauf, wer-

den aber im nächsten Heft wieder darauf zurückkommen. Wir treffen eine weitere Vereinbarung, die uns aber durch unser Computersystem eigentlich bereits aufgezwungen wird. Demnach hat dem Rufzeichen ein Anführungszeichen zu folgen, damit nicht etwa der Name des Macros tokenisiert wird. Es folgt dann der Name, welcher wiederum durch Anführungszeichen abgeschlossen wird. Weitere Befehle, ob Basic oder Macros, seien in dieser Zeile nicht mehr erlaubt. Die Macrobearbeitungsroutine überliest das für sie uninteressante Anführungszeichen und besorgt sich in einer Schleife den Macronamen. Doch was nun? Irgendwo in irgendeinem File nach irgendeiner Macrodefinition zu suchen hört sich wohl doch etwas kompliziert an. Wir überlassen die ganze Sucherei unserem Computersystem, indem wir für jedes Macro ein Macrofile hernehmen, welches wir nur eröffnen müssen, um an unser Macro heranzukommen. Anstatt für das Macro nun eine extra Lese- und Schreibroutine

FILE-SWITCHING DURCH KANALNUMMER

herzunehmen, behandeln wir es fast genauso wie unsere bereits vorher abgehandelten Basiczeilen, lediglich geringfügige Änderungen sind angebracht. Damit wir nun aus dem Macrofile lesen können, stellen wir die Kanalnummer ch=1 um auf ch=2. Indem wir unser Macroflag f1=0 auf f1=1 stellen, merken wir uns, daß nun ein Macro in Bearbeitung ist. Diesem müssen wir nämlich Rechnung tragen, wenn es um die Bestimmung der Zeilennummer geht, oder wenn das Ende des Macrofiles erreicht ist. Dann nämlich soll nicht gleich das Programm beendet, sondern lediglich der Macrofile geschlossen und mit dem Quellfile fortgefahren werden. Im letzteren Falle ist das Macroflag wieder auf f1=0 zu stellen, im ersteren Falle wird ganz einfach die vorherige Zeilennummer um 1 erhöht, für ausreichende Zwischenräume muß gesorgt werden.

UNSER PROGRAMMLISTING

Die Eingabe des Filenamens, Fileeröffnung, Kanalwahl, Nullstellen des Macroflags und Schreiben der Ladeadresse erfolgen mit:

```
100 SCNCLR
110 PRINT "MACRO-
    PORZESSOR"
    :PRINT
120 INPUT
    "QUELLFILE";QF$
130 OPEN1,8,2,Q$+
    ",P,R"
140 OPEN3,8,2,Q$+
    ",P,W"
150 CH=1:FL=0
160 PRINT#3,chr$(1);
    CHR$(16);
```

Lesen der Linkadresse mit Verzweigung bei Fileende, Bestimmung der Zeilennummer und Lesen des ersten Bytes in der aktuellen Zeile mit Sonderbehandlung bei Rufzeichen erfolgen mit:

```
170 GOSUB 350:Y$=X$
    :GOSUB 350
180 IF Y$=X$ AND
    X$=CHR$(0) THEN
    390
190 GOSUB 230:GOSUB
    350
200 IF X$="!" THEN
    280
```

Lesen der Basiczeile aus dem Quell- bzw. Macrofile erfolgt mit:

```
210 DO UNTIL X$=
    CHR$(0):L$=L$+X$
    :GOSUB 350:LOOP
```

Die Linkadresse berechnet sich folgendermaßen: Zeilenendezeichen, Linkadresse und Zeilennummer schlucken 5 Bytes, so daß zur Länge der Basiczeile, len(l$), die Zahl 5 zu addieren ist. Dieser Wert wird nun zur vorigen Linkadresse addiert und nun kann die Zerlegung in Low- und High-Byte erfolgen:



```
220 LA=LA+LEN(L$)+5
    :AH=INT(LA/256)
    AL=LA-256*AH
```

Bei einem Macro wird nicht die gelesene Zeilennummer genommen, sondern die, welche in der Macro-Routine bestimmt wurde und evtl in der Zeilennummerroutine leicht modifiziert worden ist:

```
230 IF FL THEN LN$
    =CHR$(LM):HN$
    =CHR$(HM)
```

Linkadresse, Zeilennummer, Basicanweisungen und Zeilenzeichen werden nun geschrieben mit:

```
240 PRINT#3,CHR$(AL)
    ;CHR$(AH);LN$;HN$;
    LS;CHR$(0):L$=""
    :GOTO170
```

Und es kann die nächste Basic-Zeile bearbeitet werden.
Einige Routinen sind noch zu behandeln.
Leseroutine:
Damit von verschiedenen Files gelesen werden kann, findet hierbei die Kanalvariable "ch" Verwendung'

```
350 GET#CH,X$:IF X$
    ="" THEN X$
    =CHR$(0)
360 RETURN
```

Zeilennummerroutine: (Zeilennummerincrement bei Macro)

```
370 GOSUB 350:LN$
    =X$:GOSUB 350:HN$
    =X$
380 IF FL THEN LM=
    LM+1:IF LM=256
    THEN HM=HM+1:
    LM=0
390 RETURN
```

Programmenderoutine:
Bei Macrofile mit Schließen desselben, Macroflag auf 0 und Kanalnummer auf 1. Dann geht es weiter mit der nächsten Basiczeile.
Im anderen Falle werden zwei Nullbytes statt der Linkadresse als Fileendekennung geschrieben und sowohl Quell- als auch Objektfile geschlossen.

```
400 IF FL THEN CLOSE
    2:FL=0:CH=1:GOTO
    170
410 PRINT#3,CHR$(0):
    PRINT#3,CHR$(0)
    CLOSE1:CLOSE3:
    END
```

Was noch übrig bleibt, ist die eigentliche Macrobearbeitungsroutine.
Der Macroname wird in die Variable MCS eingelesen. Zur Kontrolle drukken wir den Namen auf den Bildschirm. Nach der Fileeröffnung wird das Macroflag gesetzt und die Kanalnummer auf 2, den eröffneten Macrofile, gestellt. Die Ladeadresse interessiert nicht und wird überlesen.

```
280 MC$="":L$=""
290 GOSUB 350:GOSUB
    350
300 DO UNTIL X$=
    CHR$(34)
310 MC$=MC$+X$:
    GOSUB350:LOOP
320 PRINTMC$:OPEN
    2,8,2,MC$+",P,R"
330 FL=1:CH=2:GOSUB
    350:GOSUB 350
```

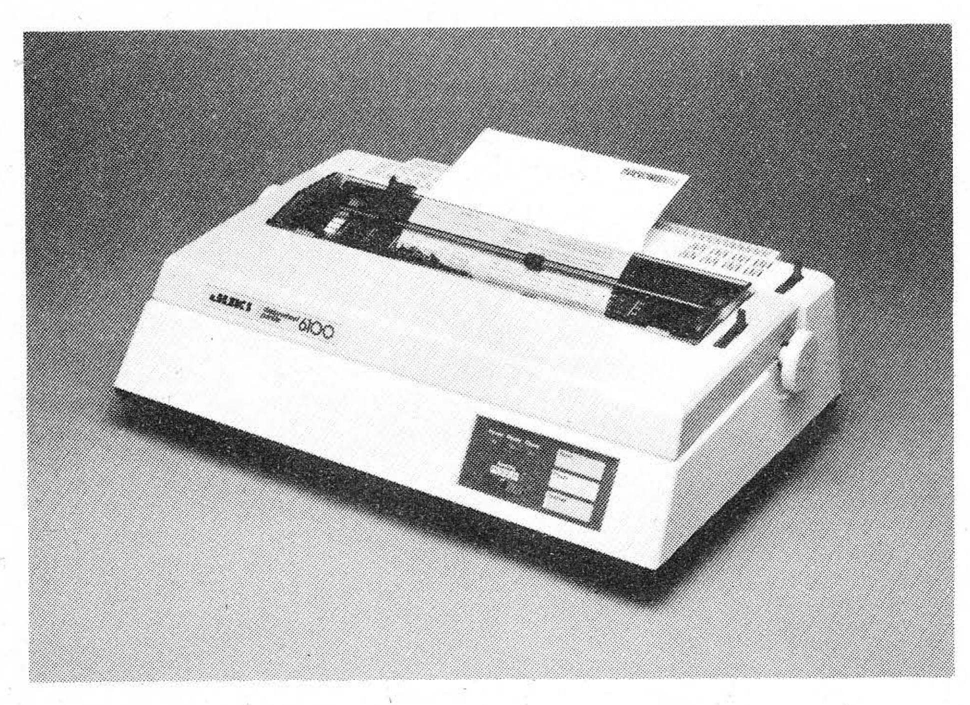
Nun werden Macrozeilennummern bestimmt und eine Remzeile mit Vermerk eingefügt. Danach geht es ganz normal weiter, mit dem Unterschied, daß nun aus dem Macrofile gelesen wird.

```
335 LM=ASC(LN$):HM
    =ASC(HN$)
340 L$=CHR$(143)+
    " "+MC$+" MACRO"
    +CHR$(0):GOTO 220
```

Sie können gleich mal experimentieren, indem Sie mit Macros wie !"GREEN", !"BLUE", oder !"RED" versuchen, Farben umzustellen.
Wenn Sie eine interessante Anwendung für Macros gefunden oder aber unser Programm ein wenig weiter ausgebaut haben, würden wir uns freuen, wenn Sie uns Ihre Ergebnisse mitteilen könnten, damit wir sie wieder der Allgemeinheit zugänglich machen können. *A. Mittelmeyer*

WENN SCHÖNHEIT ÜBE

Es gibt Anwender, die ihren Computer in Verbindung mit einem der zahlreichen ausgefeilten Programme mehr oder weniger ausschließlich zur Textverarbeitung einsetzen. Das sind die Leute, die nach wie vor entsetzt abwinken, wenn sie das Wort „Nadeldrucker" hören. Das kommt daher, daß auch der beste Nadeldrucker immer ein Matrixdrucker bleibt – auch in Korrespondenzqualität kann er nie gestochen scharfe Schönschrift liefern, die einer Schreibmaschinentype eigen ist. Darauf kommt es aber bei der Textverarbeitung sehr oft an.



Niemand rümpft die Nase, wenn die Öl- oder Gasrechnung oder auch ein Rentenbescheid mit einem Matrixdrucker zu Papier gebracht sind. Dem Leser dieser Zeilen ist es hoffentlich auch gleichgültig, daß das Manuskript auf dem Computer erstellt und mit einem Nadeldrucker in Data-Processing-Mode ausgedruckt wurde. Doch Bewerbungsschreiben und Glückwunschbriefe, Doktorarbeiten, Schriftsätze an das Gericht, Angebote, Geschäfts- oder auch Liebesbriefe – alle diese Schriftstücke können zwar auf dem Computer mit dem Textprogramm XY hergestellt werden, aber man darf es ihnen nicht ansehen. Der Absender könnte Nachteile haben.

Hier liegt die Domäne des Typenraddruckers. Er ist nämlich eine richtiggehende Schreibmaschine – und eine der technisch höchstentwickelten dazu. Das Typenrad – eine Weiterentwicklung der Typenhebel und später des Kugelkopfes – ist auch das Herzstück modernster Schreibmaschinen. Das Typenrad liefert ein kompaktes, also nicht gerastertes Druckbild, sauber und konturenscharf, aus – buchstäblich – einem Guß.

Deshalb ist es jedem Matrixdrucker in der Schriftqualität eindeutig überlegen. Insbesondere in der Schärfe, denn bei der Korrespondenz-Qualität wird zwar beim Nadeldrucker die Punktmatrix weitgehend unterdrückt, aber das erfolgt über einen technischen Trick (zweimaliges Drucken jedes Zeichens), der die Abbildung „unscharf" macht.

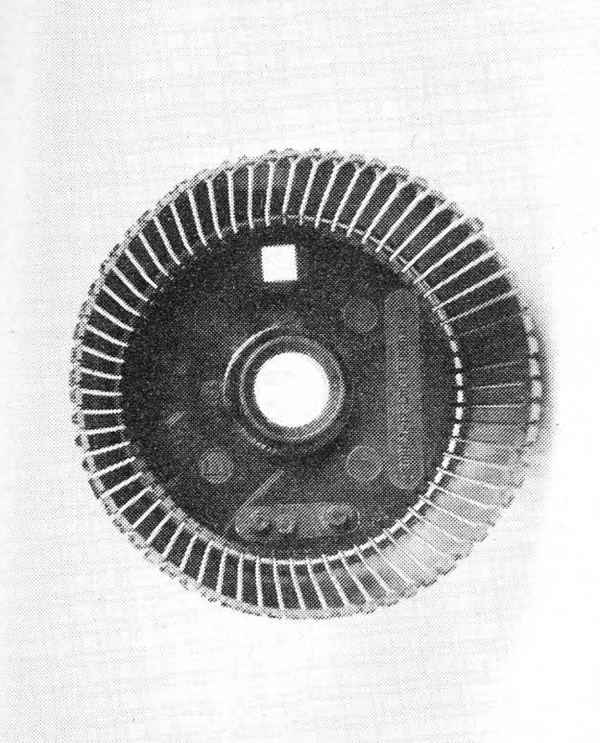
Der Typenraddrucker kennt da keine Konzessionen, und was ihn von der Schreibmaschine unterscheidet, ist allein das Fehlen der Tastatur. Der zu schreibende Text wird eben über den Computer eingegeben, die Informationen gelangen aus dessen Speicher über den Print-Befehl in den Drucker. Der liefert immer und in jedem Fall „Schönschrift", „Korrespondenzqualität". Anders kann das Typenrad gar nicht.

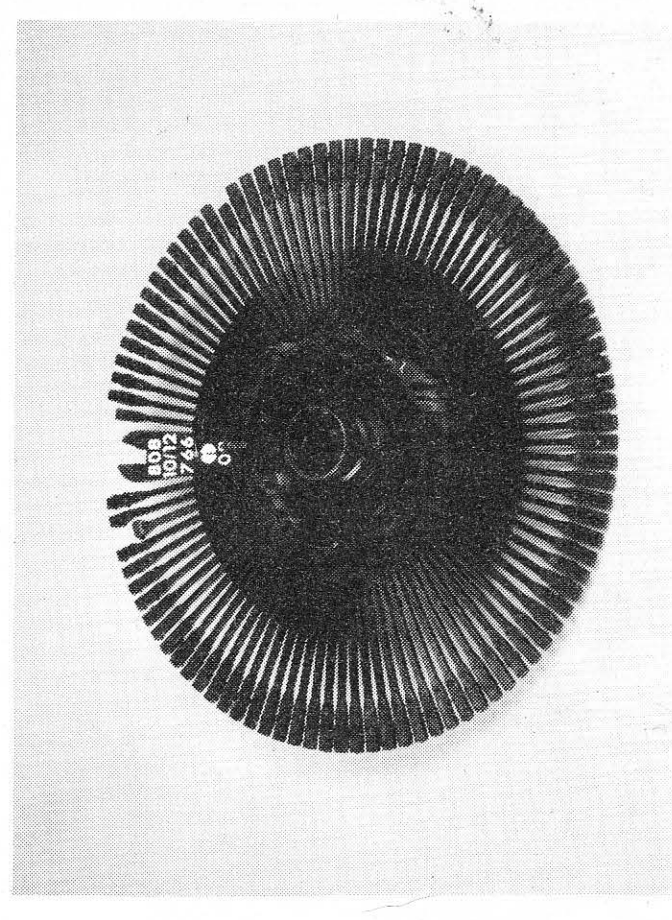
Einige andere Dinge kann es jedoch. Mit dem Typenrad sind fette und halbfette Schriften ebenso möglich wie Hervorhebungen durch automatische Unterstreichung. Je nach Modell sind auch hoch- oder tiefgestellte Zeichen möglich. Zeichenabstände im 10er- oder 12er Schritt lassen sich ebenso einstellen wie die Zeilenabstände – alles Eigenschaften, die den Typenraddrucker in einer gewissen Weise dem Matrixdrucker annähern. Verschiedene Buchstabenbreiten lassen sich nicht erzeugen. Dafür aber sind die verschiedenen Schrifttypen von der schreibmaschinentypischen Pica über die Elite und die „vornehme" Courier bis hin zu Sonderschriften wie Gothic und der frakturähnlichen Bold oder der großen, für Preisschilder und Aushänge besonders geeigneten Orator und nicht zuletzt für Proportionalschriften wesentlich leichter als beim Matrixdrucker zu wechseln. Der Matrixdrucker verlangt ein Umprogrammieren auf der Hard- und/oder Softwareseite; beim Typenraddrucker wird mit einem Handgriff das Typenrad gewechselt – fertig.

Hier ist ein Punkt, auf den man schon vor der Anschaffung des Druckers achten muß: Typenräder mit weitgehend gleichen Eigenschaften haben höchst unterschiedliche Preise zwischen – orientiert an der Liste eines Händlers – 19.– und 119,– DM, je nach Kompatibilität. Es gibt Diablo-, Olivetti-, Triumph-Adler-, Qume- und Xerox-Diablo-kompatible Typenräder. Auf den jeweiligen Drucker, gleich welchen Herstellers, paßt immer nur eine Typenrad-Serie. In dieser Hinsicht sollte man sich die Auswahl an Typenrädern anschauen – auch die ist höchst unterschiedlich.

Typenraddrucker sind heute

R ALLEM STEHEN SOLL





kaum noch teurer als leistungsfähige Nadeldrucker. Dieser Umstand könnte zu der Überlegung führen, sich doch gleich einen Typenraddrucker zu kaufen.

Da gibt es einen Haken: Typenraddrucker sind erheblich langsamer als Matrixdrucker. Zwar nicht mehr so langsam wie in ihren ersten Jahren, als sie noch mit 15 bis 20 Zeichen pro Sekunde arbeiteten, aber bei etwa 40 bis 60 cps liegt der Leistungsdurchschnitt. Das bedeutet, daß sie für die Datenverarbeitung doch zu langsam sind. Außerdem sind sie nicht graphikfähig. Zudem sind sie nicht gerade leise. Sie erzeugen ein typisches Schreibmaschinenklappern.

Standardwerkzeug

Man muß hier noch einmal sagen, daß der Nadeldrucker eigentlich das „Standardwerkzeug" neben dem Computer und unersetzlich ist, sofern der Anwender neben der Text- auch Datenverarbeitung betreibt. Tut man beides, so dürfte man nicht umhin können, sich entweder doch für den Matrixdrucker mit Schönschreibfähigkeiten oder aber für beides zu entscheiden. Die doppelte Ausgabe für die Anschaffung kann sich durchaus – und auch für den reinen Textverarbeiter

Erst Probe-Druck per Nadel, dann die Schönschrift

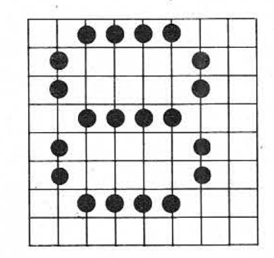
– bezahlt machen. Längere Texte, die noch überarbeitet werden müssen, lassen sich auf dem Matrixdrucker in Datenqualität und auf Endlospapier blitzschnell und farbbandsparend ausdrucken. Nach der letzten Überarbeitung, wenn die „Reinschrift" fällig ist, kommt dann der Typenraddrucker zum Zuge. Zu diesem Zeitpunkt kann dieser wieder wirtschaftlicher sein. Während nämlich der Nadeldrucker im (doppelt gedruckten) Schönschreibmodus seine meist teuren, nur bei wenigen EDV-Fachhändlern vorrätigen Spezialfarbbänder geradezu frißt und bei sehr umfangreichen Texten ein dennoch immer blasser werdendes Schriftbild liefert, bleibt beim in den meisten Schreibwarengeschäften erhältlichen Kohleband des Typenraddruckers das Schriftbild bis zum letzten Band-Zentimeter gleichmäßig „satt". Müssen mehr als drei Durchschläge gemacht werden, so ist auch da der Typenraddrucker im Vorteil – der Nadeldrucker schafft in einem Druckvorgang allerhöchstens drei Kopien. Für weitere ist ein neuer bandfressender Durchgang notwendig.

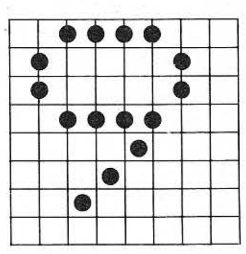
Wer viel „Schönschreiben" muß und seine Korrekturen mehr auf dem Bildschirm als auf Papier macht, findet im Typenraddrucker die bessere Wahl. In allen anderen Fällen muß sich der Anwender zumindest für die Priorität entscheiden.

Wie beim Auto, wo es auch keinen schwimmfähigen Geländewagen mit höchstem Autobahn-Fahrkomfort gibt.

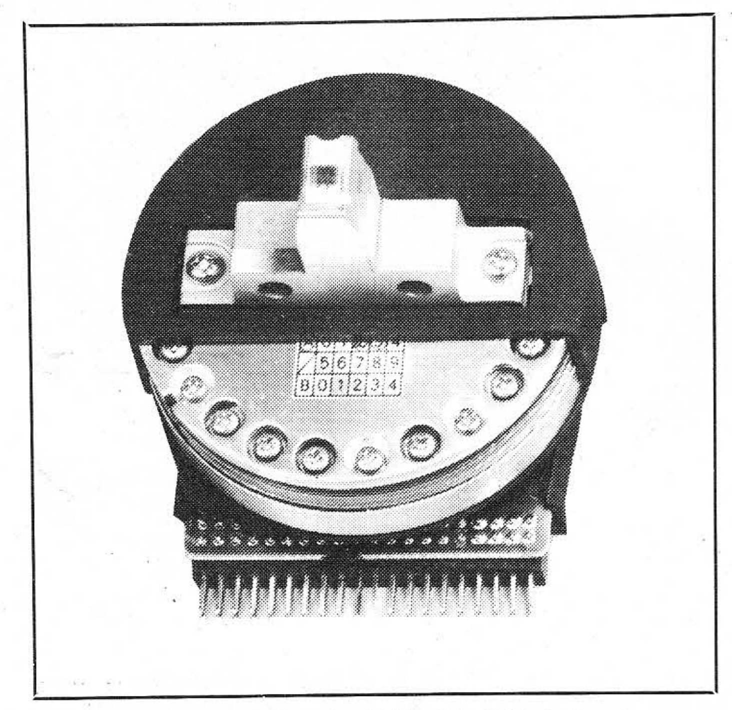
Nadeldrucker:

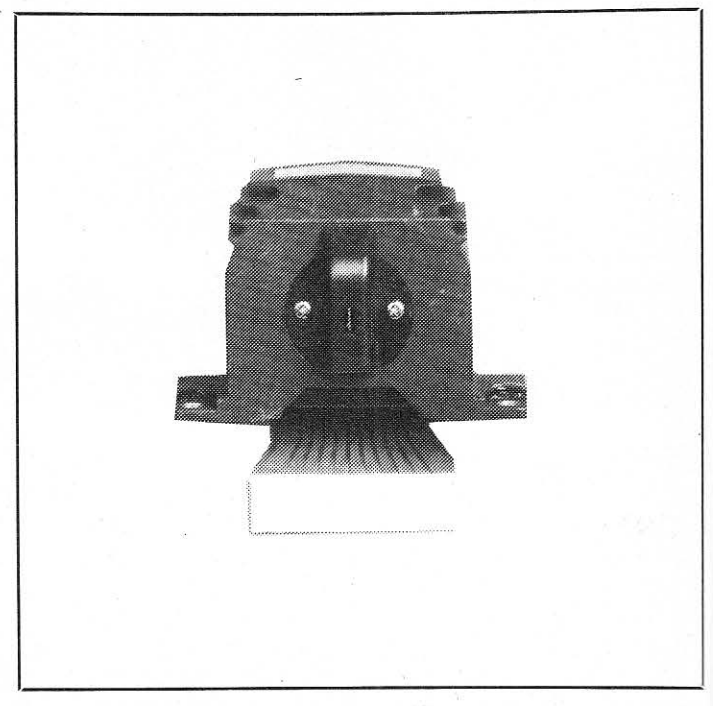
DAS GUTE STÜCK FÜR ALLE COMPUTER





Es gibt Leute, die beinahe entsetzt abwinken, wenn sie das Wort „Nadeldrucker" hören. Das kommt daher, daß dieser Druckertyp einen schlechten Ruf, ein „Vorleben" hat. Aus seinen Flegeljahren, als er unbändig laut war und miserabel schrieb, jedes Zeichen aus im Schriftbild deutlich sichtbaren Punkten zusammensetzte. Inzwischen ist der Nadeldrucker erwachsen geworden – und kultiviert dazu. Doch das Vorurteil gegen ihn ist vielfach geblieben. Völlig zu unrecht. Man muß sogar sagen, daß der Nadeldrucker eigentlich das „Standardwerkzeug"´ neben dem Computer und unersetzlich ist. Keiner kann so viel wie er und kein anderer kann es so billig.





Der Druckkopf des Fujitsu hat 24 Nadeln (links), der eines „normalen" Druckers 9 (rechts). Das ergibt schon Qualitätsunterschiede.

Der Nadeldrucker ist ein sogenannter Matrixdrukker. Diese Eigenschaft hat er mit dem Tintenstrahldrucker gemeinsam. In beiden Fällen werden die darzustellenden Zeichen aus Einzelpunkten zusammengesetzt. Stellt man sich eine 5x7 Matrix vor, so ist das ein Muster, das aus sieben senkrecht angeordneten Punkten besteht, von denen jeder in fünf waagerechten Positionen angebracht werden kann. Betrachten wir uns ein L und ein E in dieser Matrix.

Im Prinzip also ganz einfach. In den Anfangszeiten des Nadeldruckers hat man es aus technischen Gründen etwas zu einfach gemacht – daher kam die sehr schlechte Darstellung komplizierter Zeichen wie etwa des R oder einer 8. Ursprünglich schaffte ein 5x5 Matrixdrucker nur Großbuchstaben. Ein „g" etwa war gar nicht zu bewerkstelligen.

Heute sind die 7x7- oder 9x9-Matrix der Standard auch schon in den niedrigsten Preisklassen; jedes Zeichen wird also von sieben oder neun waagerechten Positionen gebildet. Aus 49 oder 81 Punkten aber läßt sich jedes nur denkbare Zeichen gut darstellen. Sogar die Sonderzeichen sind kein Problem mehr.

Sonderzeichen kein Problem

Im Druckkopf des Nadeldruckers befinden sich, wie der Name sagt, der Matrix entsprechend sieben, neun oder auch noch mehr „Nadeln" – sehr feine, ungemein präzise gearbeitete Stahlstifte, die für jedes abzubildende Zeichen vom Mikroprozessor des Druckers (der seine Informationen wiederum über den Computer erhält) einzeln angesteuert werden. Haben die gemeinsam wirkenden Chips im Computer und im Drukker die Nadeln so gesteuert, daß die gewünsch-

te „Abbildung“ (nichts anders ist ja ein Buchstabe, eine Zahl, ein Sonderzeichen) entsteht, dann werden die benötigten Nadeln per magnetischer Kraft aus dem Druckkopf „herausgeschossen“, treffen auf das Farbband und pressen dieses in der gewünschten Form auf das Papier. Das alles passiert in unvorstellbar kurzer Zeit – bei einer 7x7-Matrix und 160 Zeichen pro Sekunde sage und schreibe je nach Zeichen bis zu 7840 mal in einer Sekunde. Und jedesmal muß die zu jeder Nadel gehörende Kraftfeld-Spule wieder aufgeladen werden. Deshalb wird der Druckkopf unter der Arbeit so heiß, daß er bei jedem Nadeldrucker mit einer entsprechenden Warnung versehen ist. Man kann sich sonst ganz schön die Finger verbrennen. Manchmal wird der Druckkopf überhitzt. Man merkt es daran, daß der Drucker dann eine Pause macht, ein paar Sekunden lang. Dem Anwender fällt es kaum auf – Pausen macht der Drucker auch, wenn er sich aus dem Computer weiteren Text in seinen Pufferspeicher holt.

Das Prinzip der einzeln angesteuerten Nadeln macht den Nadeldrucker so vielseitig. Bei den heute angebotenen Modellen sind durchweg nicht nur mehrere Zeichensätze wählbar (also die Schriftarten Elite, Pica, Kursiv), sondern auch mehrere Schriftgrößen und alle nur denkbaren nationalen Zeichensätze wie etwa die deutschen Umlaute und das ß, die durchstrichenen skandinavischen Zeichen oder das spanische kopfstehende Fragezeichen. Meist läßt sich der Druck in der sehr schnellen Datenqualität oder der langsameren, aber dafür gestochen scharfen Korrespondenz (Near Letter)-Qualität wiedergeben. Selbst im letzten Fall sind die Nadeldrucker von heute so schnell, wie sie es früher in ihrer Primitivschrift waren. Das wurde mit einem einfach anmutenden, aber doch genialen Kunstgriff erreicht, dem bidirektionalen Druck: Nicht nur auf dem normalen Weg des Druckkopfes von links nach rechts (wie jede Schreibmaschine es macht) wird gedruckt, sondern auch auf dem Rückweg. Das heißt, daß der oben geschilderte Vorgang des Zeichenbildens dann umgekehrt ablaufen muß. Das aber machen die Drucker „druckwegoptimiert“ – sie drucken die jeweils nächste Zeile in der Richtung, für die der Kopf den kürzesten Weg zurücklegen muß.

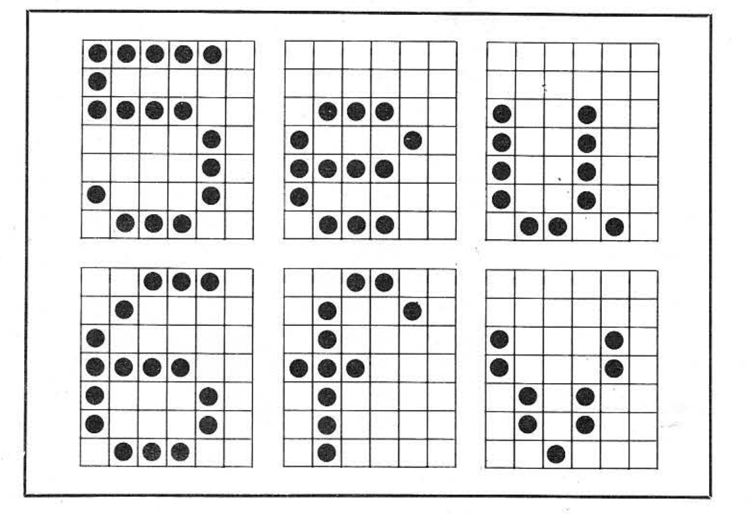
Man kann hier von einem technischen Wunderwerk sprechen, das den Anwender manchmal mehr beeindruckt als der Computer selbst.

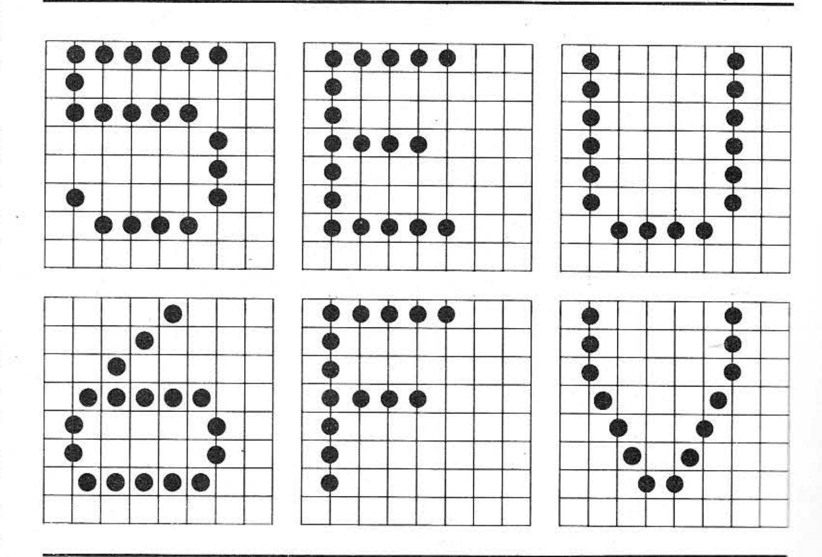
Dazu kommt noch, daß die Drucker heute viel leiser arbeiten als früher. Nicht gerade lautlos, aber doch erträglich, leiser als eine Schreibmaschine.

Wie gesagt, der Nadeldrucker kann unglaublich viel. Von der lupenkleinen Schrift bis zur Schlagzeile. Normale Zeilen, komprimierte. Normaler Zeichenabstand, enger. Dazu noch Blockgrafik. Im Normalfall 100 bis rund 200 Zeichen pro Sekunde schnell.

Dabei ist er sparsam. Fast alle modernen Nadeldrucker sind so ausgestattet, daß sie sowohl normale Einzelblätter jeder beliebigen Stärke vom Durchschlagpapier bis zum Postkartenkarton als auch Endlospapier mit Traktorführung verarbeiten. Papiersalat bei letzterem gibt es auch kaum noch, seitdem die Drucker mit einer Papierzuführung von der Druckerunterseite ausgestattet sind: Man kann den Drucker auf den Papierkarton stellen, das beschriebene Papier kann ohne Berührung mit dem zulaufenden Papier abgeführt werden. Letzter, nicht zu unterschätzender Vorteil: Genau wie auf einer Schreibmaschine kann man Durchschläge herstellen.

Womit der Nadeldrucker wirklich ein Universalgerät ist. Es gibt kaum einen Computeranwender, der ihn nicht voll einsetzen könnte und kaum ein Einsatzgebiet, auf dem er nicht zufriedenstellend arbeiten würde.





Eine gute Computerzeitschrift steigt und fällt mit der Qualität ihrer Leser. Ohne die vielen wertvollen Beiträge, die uns in Form von Postkarten, Briefen, aber aber auch auf Datenträger erreichen, wären wir wohl stark überlastet. Woher sollten wir all die kleinen aber ungemein wirkungsvollen Tips und Tricks, die Superspielepokes und so weiter erhalten, wenn nicht von unseren Lesern ? Jeder unserer Käufer ist dazu aufgefordert, uns seine Erfahrungen mit dem Computer oder mit seinem Händler per Brief zuzusenden. Programme allein füllen eine Zeitschrift nicht. Sicherlich gehören Sie unbedingt dazu, ist es doch um einiges einfacher, die Programmiersprache BASIC gleich an Hand eines Beispieles zu erlernen als ein trockenes Handbuch durchzublättern. Dennoch müssen ebenfalls andere Informationen weitergegeben werden.

Nehmen wir zum Beispiel die Clubs : Durch die Möglichkeit, daß sich jeder Computerverein bei uns vorstellen und seine Tätigkeiten beschreiben kann, werden über die COMMODORE WELT Kontakte geknüpft, die jedem Teil Vorteile bringt:
- Dem Club, da er seine Miztgliederzahl erhöhen kann und somit effizienter wird.
- Dem Leser, der sich mit seinen Problemen an fachkundige Spezialisten wenden kann
- Der Zeitschrift, die durch diese Vermittlungen erstens Leser in entscheidender Weise hilft, was diese meist zweitens durch den fortwährenden Kauf unseres Heftes honorieren.

Überdies haben wir immer die Hoffnung, daß die verschiedenen Computerclubs uns ihre verschiedenen Infos zur Verfügung stellen, so daß die Redaktion stets auf dem neuesten Stand der Dinge ist.

Dennoch gibt es mit verschiedenen Leser - und Clubzuschriften immer wieder kleine und große, aber vor allem nicht notwendige, Probleme.

Um die Mitarbeit unserer Leser zu optimieren, wollen wir auf die häufigsten Fehler zu sprechen kommen. Beginnen wir mit:

- PROGRAMMEINSENDUNGEN

Programmformat : Täglich bekommen wir aus allen Teilen Europas Leserprogramme für unsere unterschiedliche Geräte.

Das ist erfreulich.

Täglich versenden wir Dutzende von Leserprogrammen, weil diese nicht unserem Format entsprechen.

Das ist weniger erfreulich.

Doch dagegen können Sie etwas tun: Bevor Sie uns ein Programm (oder vielleicht auch mehrere) zusenden, nehmen Sie sich eines unserer Hefte und schauen sich die dort abgedruckten Listings für Ihren Rechnertyp an.

Sicher haben Sie schon festgestellt, daß wir die Commodore-spezifischen Invers - und Grafikzeichen vermieden und statt dessen Variable eingesetzt haben. Programme unserer Leser, die diesen Schritt ebenfalls unternommen haben, werden auf jeden Fall näher betrachtet, Programme, die noch modifiziert werden müssen, werden an den jeweiligen Autor zurückgeschickt, der dann mit dieser Aufgabe betraut wird. Vorteilhaft für jeden Programmeinsender ist, wenn er seine Programme mit dem Copyrigthkopf und Programmschluß aus stattet, den wir ebenfalls benutzen.

Nur selten haben wir noch genügend Zeit, um uns mit der Modifikation der Programme zu beschäftigen, so daß nur die wirklich allerbeste Software von uns geändert wird.

Programmbeschreibung :

Mittlerweile sollte eigentlich bekannt sein, daß auch einem Programm, das sich großteils selbst erläutert, eine kurze, ca. 1 seitige, Beschreibung beiliegen sollte. Erstens wissen wir dann schon im Voraus, um was es sich bei dem Programm handelt und zweitens veröffentlichen wir die Beschreibung gleich mit, so daß der Programmeinsender ein höheres Honorar zu erwarten hat.



Datenträger:

Haben Sie schon einmal einen Disketten-ID-Changer auf Kassette erhalten ?

Wir schon ! Und das fanden wir gar nicht gut. Denn wer ein Laufwerk besitzt, was wir bei einem solchen Programm voraussetzen, der kann uns auch eine Diskette senden, die wir schneller laden und testen können als eine Kassette.

Daher : Wer eine Floppy hat, sollte uns immer eine DISKETTE und keine Kassette zusenden. Datasettenbenutzer senden uns selbstverständlich weiterhin ihre Kassetten.

Programme :

Von allen Programmen, die wir im Laufe einer Woche auf unserem Schreibtisch liegen haben, entfallen bestimmt 60 - 70 Prozent auf:

Lottozahlenermittlung
Vokabeltrainer
Mathetrainer
Würfelspiele
Datenbanken (Adressverwaltung, ...)
und Kartenspiele

Wir können allerdings nicht in jedem Heft jeweils eines dieser Programme veröffentlichen, so daß wir einen erheblichen Mangel an Programmen anderer Sparten verzeichnen müssen.

Wie wäre es denn mit einem kleinen Ballerspiel gekoppelt mit etwas Geschicklichkeit und Raffinesse ? Oder einem (wirklich) einfallsreichen Adventure mit mehr als 5 Befehlen und 3 Räumen ?

Es gibt doch noch derart viele Möglichkeiten, die der C64 bereithält. Warum sollten DIE nicht ausgenutzt werden ?!?

Ausgenutzt werden sollte ebenfalls die Möglichkeit, sich schriftlich zu äußern.

- ARTIKEL UND LESERBRIEFE

Leserbriefe erfordern keine besondere Form und können daher auch handschriftlich an uns geschickt werden. Artikel hingegen müssen das Format von 40 Zeichen Breite und 60 Zeilen Länge je Seite beinhalten.

Damit haben wir die Möglichkeit, das Schreiben nach unseren Vorstellungen, falls notwendig, zu modifizieren. Optimal erscheint uns aber die Möglichkeit, daß uns der Artikel sowohl als Ausdruck als auch auf einem Datenträger zugesandt wird.

Da wir so ziemlich jedes Textverarbeitungsprogramm für die Commodore Maschinen in unserem Archiv haben, können wir die auf der Diskette befindlichen Daten in optimaler Weise korrigieren und in unserem Format ausdrucken bzw. auf einer Satzmaschine belichten

Ein weiteres Problem scheint die Honorarfrage zu sein:

- HONORAR

Hat sich bei uns ein Redakteur für die Veröffentlichung eines Programmes oder eines Artikels entschieden, so wird der betreffende Autor umgehend benachrichtigt. Danach wird das Werk in eines der nächsten Hefte übernommen. Damit ist der Autor honorarberechtigt. Dies erhält er allerdings nicht gleich nach einer Veröffentlichung, sondern erst nach einer angemessenen Frist, damit auch die Kassetten bzw. Diskettenbestellungen der Leser in die Honorierung mit einbezogen werden können.

- RÜCKPORTO

Manuskripte und Datenträger, die unaufgefordert eingesandt wurden, sollten mit Rückporto versehen sein oder können Sie sich vorstellen, welche Kosten uns ansonsten enstehen ?!?

- CLUBECKE :

Wollen Sie einen Club gründen oder suchen Sie weitere Mitglieder in Ihrer Umgebung? Senden Sie uns bitte, falls vorhanden, ein Clubinfo und vor allem genaueste Informationen über :

Verwendeter Computer
Beitragsgebühren
Treffen (wann, wo, wie oft).
Mitgliederanzahl
Clubsatzung
Hauptthemen des Clubs

Nur so können wir unsere Leser umfassend über die Aktivitäten der unterschiedlichen Computerclubs informieren.

Zum Schluß möchten wir nochmals die Gelegenheit wahrnehmen, alle Leser unserer Zeitschrift aufzufordern, sich an ihren Computer zu setzen und uns die Produkte ihrer geistigen Arbeit zuzusenden.

Gerne teilen wir Ihnen bei einer Anfrage mit, welche Programme wir gerade benötigen und welche im Überfluß vorhanden sind.

Die Adresse steht auf unserem Blatt "Verdienen Sie Geld mit Ihrem Computer".

DIE WICHTIGSTEN GRAFIK-BEFEHLE

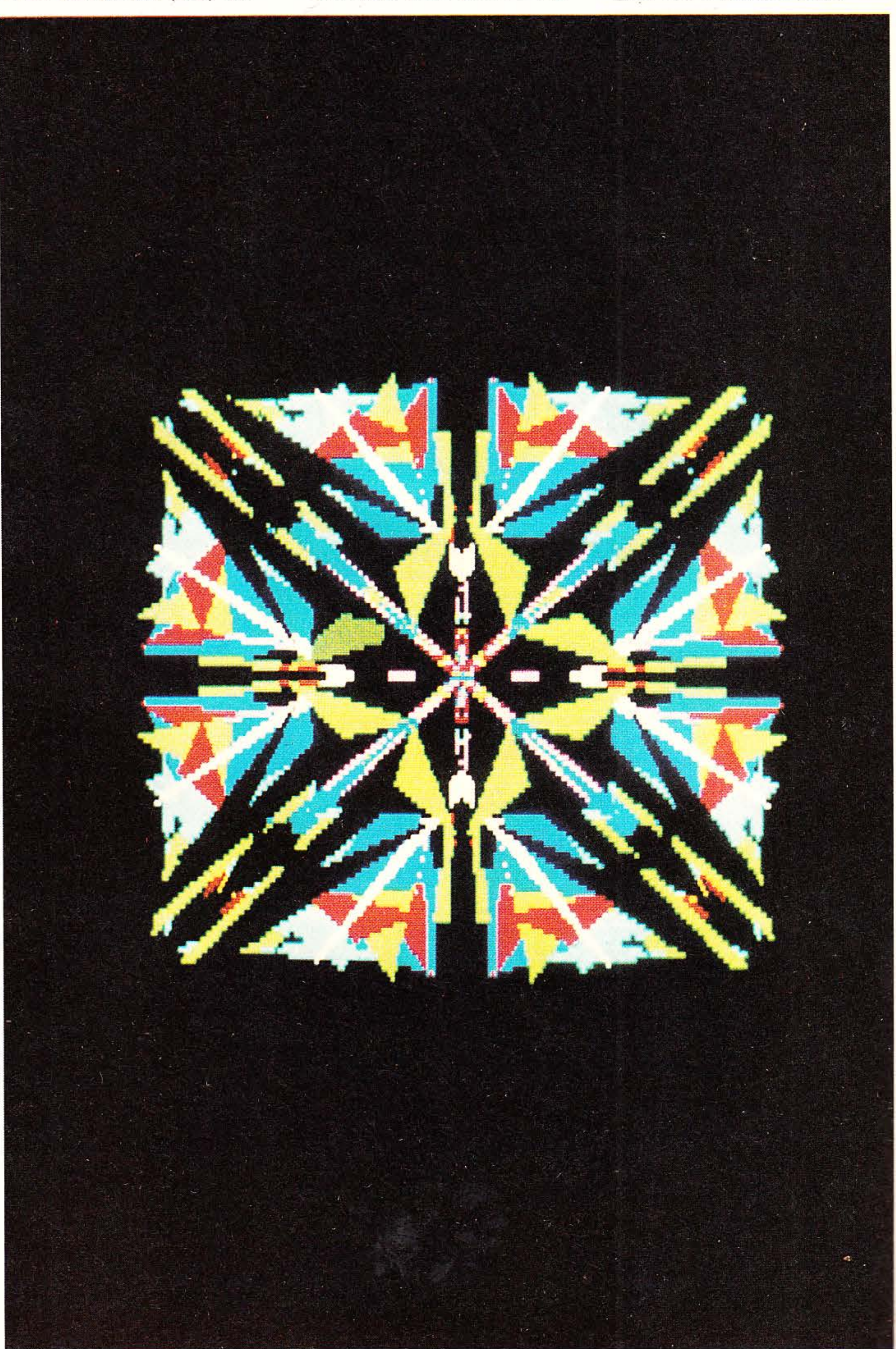
Dieser Artikel soll nicht das Handbuch ersetzen, sondern solche Befehle näher behandeln, die aufgrund ihrer Vielseitigkeit etwas kompliziert sind und so vom ungeübten Programmierer nicht voll genutzt werden können.

Die hochauflösende Grafik des C16 mit einer Auflösung von 320x200=64000 Punkten in 16 Farben und 8 Helligkeitsstufen ist eine seiner hervorragendsten Eigenschaften. Doch erst die starken Grafik-Befehle ermöglichen eine optimale Ausnutzung dieser Fähigkeiten. Leider steht dem C16 in der Grundversion nur noch etwas mehr als 2 KByte Speicher zur Verfügung. Es ist aber erstaunlich, was man damit alles anfangen kann, besonders, wenn man die Möglichkeit des Nachladens von weiteren Programmteilen nutzt. Wie man in bestimmten Fällen auch den Bildschirmspeicher für Programme nutzen kann, wurde in der COMMODORE WELT 2/87 erklärt.

Die hochauflösende Grafik wird im allgemeinen nicht für Spiele eingesetzt, da hierzu der Bildschirmaufbau zu langsam ist. (Für Spiele verwendet man einen veränderten Zeichensatz.) Das Einsatzgebiet liegt eher bei der Darstellung von Kurven und Bildern zur Verdeutlichung von mathematischen oder statistischen Zusammenhängen und der künstlerischen Grafik. In diesem Abschnitt sollen nun die beiden umfangreichsten Grafikbefehle, CIRCLE und SSHAPE, behandelt werden. Um alles verstehen zu können, muß jedoch noch der Begriff des Pixelcursors erklärt werden.

DER PIXELCURSOR

So wie im Textmodus der blinkende Rechteckcursor die Position anzeigt, bei der das nächste Zeichen ausgegeben werden soll, genauso enthält der Pixelcursor (PC) die Position (und die Farbe) des zuvor gezeichneten Punktes. Man kann diese Werte abfragen:
print rdot(0)
gibt die x-Position an



```
print rdot(1)
```
gibt die y-Position an
```
print rdot(2)
```
gibt die Farbzone an. Diese Werte können beim Zeichnen als Startwerte genutzt werden. Gibt man zum Beispiel den Befehl:
```
draw to 100,100
```
ein, so wird eine Linie von der derzeitigen PC-Position bis zu x=100 und y=100 gezeichnet. Vor 'to' steht übrigens *nie* ein Komma, wie es an manchen Stellen im Handbuch auftaucht.

Besonders interessant ist in diesem Zusammenhang die relative Bewegung, indem man anstelle des Kommas einen Strichpunkt benutzt:
```
draw to 100;60
```
zeichnet eine Linie von der derzeitigen PC-Position mit einer Länge von 100 unter dem Winkel von 60 Grad.

Eine weitere Relativbewegung wird mit +/– ausgeführt:
```
draw to +100,–100
```
zeichnet eine Linie von der PC-Position bis nach x=rdot(0)+100 und y=rdot(1)–100.

In jedem Fall wird mit der Farbe rdot(2) gezeichnet.

Die Lage des Pixelcursors kann man mit LOCATE x,y festlegen, die Farbe (F) und die Luminanz (L) mit COLOR 1,F,L.

Die Vielzahl der Parameter, die dem CIRCLE-Befehl übergeben werden können, zeigt schon, daß dieser sehr vielseitig ist. Wie Sie sicher aus dem Handbuch wissen, können damit nicht nur Kreise (circles) sondern auch Kreisausschnitte, Polygone und Ellipsen gezeichnet werden.

Da der beste Weg, das Programmieren zu erlernen, das Programmieren selbst ist, schalten Sie doch bitte einmal mit GRAPHIC 2,1 die hochauflösende Grafik mit Textfenster ein, geben die folgenden Beispiele ein und variieren Sie nach Belieben. So bekommen Sie rasch einen Überblick über die vorhandenen Möglichkeiten.

Zur Eingabe einige Tips:

Durch GRAPHIC 2,1 werden HIRES- und Textbildschirm gelöscht, der Cursor befindet sich im Textfenster. Bei GRAPHIC 2 bleibt der Cursor jedoch am alten Platz. Man muß dann mit dem Cursor nach unten fahren. Nachdem ein Befehl ausgeführt wurde, verschwindet die Befehlszeile oft nach oben. Will man in der gleichen Zeile Änderungen vornehmen und sie dann nochmal ausführen lassen, kann man sie mit ESC&W (ev. mehrmals) wieder nach unten holen.

Das Beispiel-Programm 'CIRCLE' enthält alle die folgenden Beispiele und noch einige mehr, wobei die entsprechende BASIC-Zeile zur Kontrolle angezeigt wird.

DER CIRCLE-BEFEHL

Schreibweise:

CIRCLE (fz),(x,y),xr,(yr),(wa),(we),(dw),(inc)

		Normalwerte (beim Weglassen)
fz	Farbzahl (0–3)	1
x,y	Mittelpunktskoordinaten	rdot(0),rdot(1)
xr	X-Radius	xr
yr	Y-Radius	yr
wa	Anfangswinkel (in Grad)	0
we	Endwinkel (in Grad)	360
dw	dreht Figur um X Grad	0
inc	Winkel zwischen 2 Segmenten	2

```
circle,160,100,50
```
zeichnet einen Kreis um x=160,y=100 mit dem Radius 50; yr ist ebenfalls 50, da nichts angegeben wurde. Steht vor dem ersten Komma 0, so wird der Kreis gelöscht.

Auch für x und y sind Relativwerte möglich und können viel Rechenaufwand ersparen:
```
for w=0to360 step 20:
locate160,100:circle1,
40;w,10:next
```
zeichnet 18 Kreise mit Radius 10, deren Mittelpunkt immer 40 Punkte unter dem Winkel w vom gemeinsamen Zentrum in 160,100 entfernt sind.
```
for yr=10to100 step10:
circle,160,100,50,yr:
next
```
Hier wird der y-Radius variiert, es entstehen also lauter Ellipsen.
```
for wa=0to270 step45:
circle,+40,100,18,40,
wa:next
```
Der Anfangswinkel der Ellipsen wird variiert. Gleichzeitig wird der x-Wert des Mittelpunktes immer um 40 Punkte nach rechts verschoben (nach GRAPHIC 2,1 steht der PC bei 0,0). Beachten Sie, daß der Winkel immer oben (bei 12 Uhr) beginnt und im Uhrzeigersinn gezählt wird.
```
for dw=0to90 step10:
circle,160,100,40,60,,,
dw:next
```
Die Ellipsen werden um den Winkel dw gedreht.
```
for p=3to15:inc=360/p:
circle,160,100,50,,,,,inc:
getkeya$:graphic1,1:
next
```
Verändert man das Inkrement, so ergibt es einen eckigen Kreis.

Regelmäßige Vielecke erhält man, wenn 'inc' ein Teiler von 360 ist.

In diesem Beispiel erhält man Polygone vom 3-Eck bis zum 15-Eck, wobei letzteres dem Kreis schon sehr ähnlich sieht.

Will man schnell kleine Kreise zeichnen, so ist es sinnvoll, ein Inkrement größer 2 anzugeben, da bei kleinen Kreisen der Unterschied nicht auffällt, aber die Zeichengeschwindigkeit viel größer sein kann.

DER SSHAPE- UND GSHAPE-BEFEHL

Eine der besten Ideen von Commodore für den C16 sind die SHAPEs. Sie ermöglichen es, beliebige rechteckige Ausschnitte aus einem HIRES-Bild herauszunehmen, in einer Stringvariablen (a$,b$,bild$...) abzulegen und an anderer Stelle wieder darzustellen.

Schreibweise:

SSHAPE Stringvariable,xy,ya(,xe,ye)

Stringvariable	beliebiger Variablenname
xa,ya	linke obere Ecke
xe,ye	rechte untere Ecke

Statt einer Stringvariablen einen String ("schiff") zu verwenden, wie es im Handbuch beschrieben ist, geht leider nicht. Werden xe und ye nicht angegeben, so wird die Position des PC eingesetzt.

Wie Sie aus dem Handbuch sicher wissen, gibt es für die Größe der SHAPEs die Einschränkung, daß ein String nicht mehr als 255 Byte haben kann. Dabei werden die letzten 4 Byte für die Längenausdehnung in x- bzw. y-Richtung benötigt.

Will man die 255 Bytes, die zur Verfügung stehen, voll nutzen, so muß man wissen, daß in x-Richtung ein SHAPE immer byteweise erfaßt wird. Schreibt man zum Beispiel mit

graphic2,1:char,0,0,"a"

ein A in die linke obere Ecke und will dieses in einen SHAPE übernehmen, so geschieht dies mit:

SSHAPE S$,0,0,7,7

Man sollte also beachten, daß die Zählung bei 0 beginnt. Setzt man die Grenzen von 1 bis 8, so braucht man schon 8 Byte mehr. Jetzt wollen wir uns einmal den Inhalt von SS ansehen:

```
graphik 0
for i=0 to len(a$):print
asc(mid$(a$,i,l)):next
```

Man erhält 12 Werte. Die letzten 4 Zahlen sind die x- und y-Ausdehnung in LO/HI-Byte-Darstellung. Den umgekehrten Weg kann man beschreiten, wenn man einen SHAPE aus Zahlen zusammensetzen will. Wiederholen Sie einmal das ganze, wobei Sie im SSHAPE-Kommando ',7,7' durch ',8,7' ersetzen. Jetzt werden 20 Byte benötigt, obwohl wir in x-Richtung nur 1 Bit mehr verlangt haben. Es werden aber immer 8 Bit nebeneinander zu einem Byte zusammengefaßt. Dies führt uns zu dem Problem, wie groß ein SHAPE sein kann. Im Handbuch ist dazu eine recht komplizierte Formel angegeben. Hier ist ein kleines Programm, mit dem die Endwerte eingegeben werden können. Wird das SHAPE zu groß, so wird der ye-Wert solange reduziert, bis kein 'STRING TO LONG ERROR' mehr erzeugt wird. Diese Fehlermeldung wird von TRAP abgefangen und das Programm fortgesetzt.

```
200 input xe,ye;xe,ye
210 trap230
220 sshapea$,0,0,xe,
    ye:printxe,ye:clr
    :goto200
230 ye=ye-1:resume210
```

Damit Sie nun nicht jedesmal das Programm starten müssen, wurde diese Tabelle mit obigem Programm erstellt:

TABELLE ZUR ERMITTLUNG DES MAXIMALEN ye-WERTES BEI VORGEGEBENEM xe

xe	ye	xe	ye	xe	ye
0	251	64	27	128	13
8	125	72	24	144	12
16	83	80	21	152	11
24	62	88	20	168	10
32	49	96	18	176	9
40	41	104	17	200	8
48	35	112	15	224	7
56	63	120	14	248	6

Die Tabelle wird folgendermaßen benutzt:

Sucht man zu einem gegebenen xe-Wert den maximalen ye-Wert, so sucht man in der Tabelle den nächst niedrigeren Wert und liest den dazugehörenden ye-Wert ab. Zum Beispiel kann man bei einer x-Ausdehnung von 47 Punkten eine maximale y-Ausdehnung von 41 Punkten angeben.
Hat man ein SHAPE erzeugt, dann will man es an einem bestimmten Platz darstellen. Dazu verwendet man den GSHAPE-Befehl.

Schreibweise:
GSHAPE Stringvar.
((,xa,ya),Modus)

Die Koordinaten der linken oberen Ecke (xa,ya) und der Wiedergabemodus können auch weggelassen werden. Dann werden die PC-Koordinaten und als Wiedergabemodus 0 verwendet.
Für den Wiedergabemodus gibt es fünf Möglichkeiten:

0 wie aufgenommen	
1 reverse	Zeichenfarbe und Hintergrund vertauscht
2 ODER-verknüpft	Punkte im SHAPE kommen zu den vorhandenen Punkten dazu
3 UND-verknüpft	ein Punkt des SHAPE wird nur dann gezeichnet, wenn an der Stelle schon ein Punkt war
4 EOR-verknüpft	(Exclusiv-OR), Punkte werden nur dort gezeichnet, wo entweder beim SHAPE oder beim Untergrund 1 Punkt ist

Damit Sie die Wirkung der verschiedenen Modi besser verstehen, geben Sie einmal folgendes Beispielprogramm ein:

```
10graphic2,1
20char,0,0,"shape"
30sshape a$,0,0,len
  (a$)*8-1,7
40fori=0to20:gshapea$,
  10*i,6*i:next
50getkeyw$:scnclr
60box,50,0,100,60,,1
70locate5,5:for m=0to
  4:gshapea$,,m:next
80locate55,5:for m=0to
  4:gshapea$,,m:next
```

Das Wort SHAPE wird in die linke obere Ecke geschrieben und mit SSHAPE in die Variable A$ übernommen. In Zeile 40 wird gezeigt, wie man ein Wort an beliebige Stelle auf dem HIRES-Bildschirm platzieren kann, wobei der Untergrund überschrieben wird (Modus 0).
Zeile 60 zeichnet ein ausgefülltes Rechteck und in den Zeilen 70 sowie 80 wird das Wort 'SHAPE' mit allen 5 Modi einmal auf weiße Fläche und einmal auf schwarze Fläche geschrieben.
Der interessanteste Modus ist 4 (EOR-Verknüpfung). Zum einen, weil eine Figur auf hellem Untergrund dunkel und auf dunklem Untergrund hell erscheint, somit also immer erkennbar ist, und zum zweiten, weil eine zweimalige EOR-Verknüpfung den ursprünglichen Zustand wieder herstellt. Dadurch kann man zum Beispiel eine Figur über den Bildschirm wandern lassen, ohne das Bild zu zerstören.
Da man in der Grundversion beim C16 so wenig Speicherplatz bei eingeschalteter Grafik zur Verfügung hat, muß man alle Möglichkeiten nutzen, unnötige Angaben einzusparen. Sowohl bei SSHAPE als auch bei GSHAPE kann man die PC-Position nutzen, wie beim vorherigen Beispiel in Zeile 70. Hier wird automatisch die PC-Position eingesetzt, welche sich nach der Anwendung von GSHAPE in der linken unteren Ecke des SHAPEs befindet, wodurch man mehrere SHAPEs leicht ohne Rechenaufwand untereinander darstellen kann.
Zur Verdeutlichung folgt noch ein Beispiel, wie man die PC-Position für SSHAPE und GSHAPE ausnützen kann:

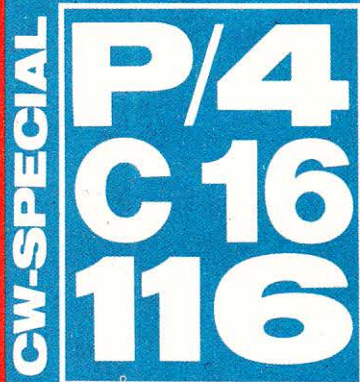
```
110 locate40,40:fori:10
    to30step10:box,i,i
    :next
120 sshapea$,10,10:draw
    to100,100:gshapea$
```

So, nun lassen Sie alle Hemmungen fahren und probieren einfach mal wild drauflos. Sie werden sehen, daß es gar nicht so schwierig ist.

Zum Abschluß noch ein kleiner Tip: Wissen Sie, wie man am schnellsten vom HIRES-Grafik-Modus in den normalen Textmodus kommt?
Mit '*RETURN'. Dies erzeugt zwar ein SYNTAX-ERROR, aber das stört ja nicht.

Software satt für C16 & P/4- User

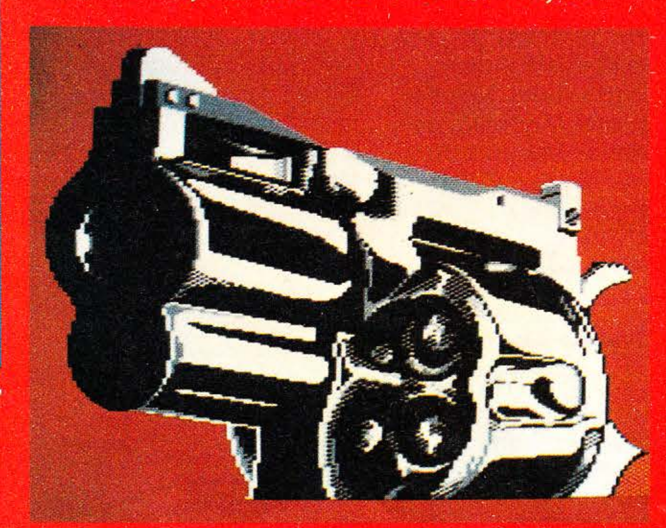




DAS BESTE AUS CW

Sammelband Nr. 1/88

DM 14,80-ÖS 124-SFR 14,80

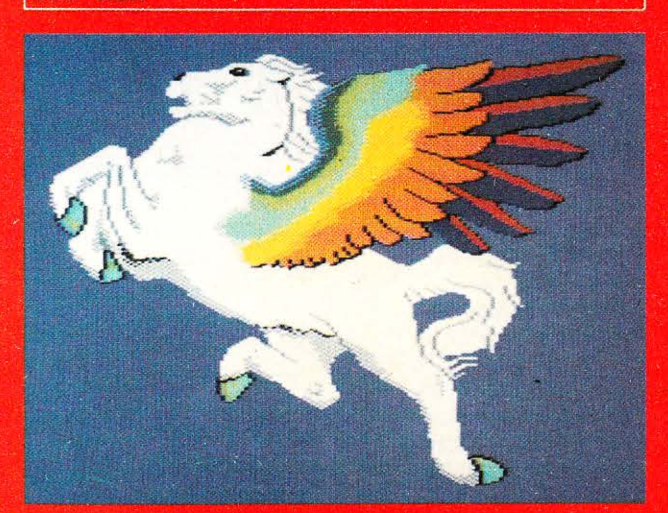


```
100 PRINT"---------------------------"
110 PRINT"THE BEST OF COMMODORE-WELT"
120 PRINT"---------------------------"
130 LOAD "LISTING 1" : RUN
140 LOAD "LISTING 2" : RUN
150 LOAD "LISTING 3" : RUN
160 LOAD "LISTING 4" : RUN
170 REM  "-------------------------"
180 REM  "IF YOU WILL LOAD AND RUN, "
190 REM  "YOU WILL HAVE A LOT OF FUN"
200 REM  "-------------------------"
```

SOFT-WARE JAHR-BUCH 1988

Super-Spiele für Ihren 16/116 und Plus 4!

Die besten Anwender-Programme



Jetzt an Ihrem Kiosk und im Bahnhofs-Buchhandel

Rund 150 Seiten! DM 14,80

WEG MIT DEN GUMMI-TASTEN! MEHR KOMFORT FÜR DIE 116-TASTATUR!

Billig ist er ja, der 116er. Aber deshalb muß der Anwender doch nicht auf den Komfort einer ordentlichen Tastatur verzichten. Dachte zumindest unser Autor. Gesagt — getan, nach etwas Tüftelei waren Schaltplan und Materialliste erstellt, hier die Anleitung zum Bau einer Profi-Tastatur für den C 16!




```
10 PRIT "S FUKTOIRT!"
20 GOT 1
RN
?SYNTAX ERROR
RUN
?SYNTAX ERROR IN 10
```

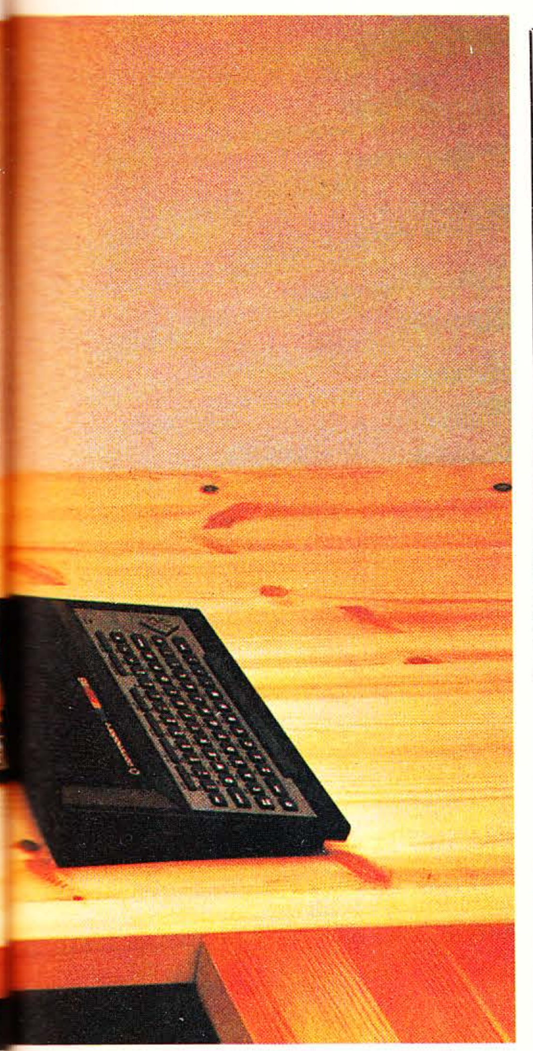
Kennen Sie das? – Ja? – Dann sind Sie sicherlich Besitzer eines „Gummitasten-Winzlings" – ich meine den C116, der seine Benutzer oft in den Wahnsinn treibt. – Schluß mit Wutausbrüchen, Fingerkuppenentzündungen, Nervenzusammenbrüchen und Fingerkrämpfen!! – Für nur ca. 40–50 DM (Materialkosten) bekommt Ihr C116 eine professionelle Tastatur mit separatem Cursor und Zehnerblock. Falls Sie Besitzer eines C16 oder Plus/4 sind und eine Tastatur mit Zehnerblock benötigen – bitte sehr: Auch für diese beiden Rechner gilt der Artikel – ebenso wie die Software – unverändert.

DAS PRINZIP

Bevor wir uns dem Aufbau des neuen Keyboards widmen, muß erst einmal geklärt werden, wie eine Computer-Tastatur eigentlich funktioniert. Wie Sie sich sicherlich denken können, besteht eine Taste aus einem Schalter, dessen Zustand („gedrückt" oder „nicht gedrückt") vom Computer erkannt werden muß.
Eine gewöhnliche Tastatur hat 60 – 90 Leitungen, die von der Tastatur zur Computerplatine führen? – Nein! – Wie durch Aufschrauben des Rechners festzustellen ist, kommt man bei den 65 Tasten (eigentlich 64, da die Shift-Tasten miteinander verbunden sind) des C116 mit 16 Leitungen aus!
Das Prinzip ist ganz einfach: Man ordnet die Tasten in einer Matrix an: Beim C16, C116, Plus/4, VC20, C64 und SX64 jeweils 8 Spalten zu je 8 Zeilen (8x8=64). Die Tastatur wird „abgefragt", indem man nacheinander auf jeweils eine Spalte eine Spannung legt, und dann „nachsieht", in welcher Zeile ein Strom fließt. Wo sich Spalte und Zeile kreuzen, muß sich die gedrückte Taste befinden, da der Schalter dort Kontakt erzeugt hat.
Zur Ansteuerung einer solchen 8x8-Matrix bedarf es also eines 8-Bit-Ausgangs für die Spalten-Auswahl (bei C116, C16 und Plus/4 der „Keyport-Baustein" 6529 an Adresse $FD30), sowie eines 8-Bit-Eingangs für die Zeilenabfrage ("TED" 8360 mit Register 8 = Adresse $FF08 – Abb. 3).

DIE TASTATURABFRAGE

Die Tastaturabfrage übernimmt eine Betriebssystem-Routine namens SCNKEY (ab Adresse $DB11); sie besorgt die Abfrage der Tastatur nach oben beschriebenem Prinzip, dekodiert die erkannte Matrix je nachdem, ob SHIFT-, COMMODORE-, CTRL- oder keine zusätzliche Taste gedrückt wurde anhand der vier Tabellen ab Adresse $E026 und schreibt den ASCII-Code der



gedrückten Taste in den Tastaturpuffer.
SCNKEY wird alle 1/60 Sekunden von der Interrupt-Routine aufgerufen, kann aber in der Kernal-Sprungliste auch über die Adresse $FF9F von eigenen Maschinensprache-Programmen aus angesprungen werden.

DIE NEUE SOFTWARE

Die von mir – weil saubillig (9,80 DM!), leistungsfähig (95 Tasten) und komfortabel (ähnlich wie bei C16/C64) – gewählte Tastatur läßt sich nicht an den Tastaturport des C116 anschließen, da sie anders belegt ist und eine andere Matrixaufteilung (16x6 statt 8x8) hat; sie wird deshalb über die in der COMMODORE WELT 10/86 auf Seite 48 beschriebene User-Port-Platine angeschlossen (Näheres siehe "Hardware").
Zwar hätte man die erforderliche neue SCNKEY-Routine über den KEYLOG-Sprung-Vektor in den RAM-Bereich legen können; da das Betriebssystem aber unkonsequenterweise die STOP- und die COMMODORE-Taste beim Laden von Kassette und beim Re-Set direkt über den Tastaturport und nicht über die SCNKEY-Routine abfragt, müßte man bei diesen beiden Tasten immer auf die alte Tastatur „umsteigen". Das Problem tritt aber bei jeder Tastatur auf, die nicht dieselbe Matrix und Tastenanordnung wie das für Commodore speziell angefertigte C16-/C116-/Plus4-Keyboard hat. Ich habe mich deshalb entschlossen, das alte Kernal-ROM durch ein modifziertes Betriebssystem-EPROM zu ersetzen, so daß die alte Tastatur völlig überflüssig wird.

	6529: SFD30 (schreiben)							
TED: SSFF08 (lesen)	0	1	2	3	4	5	6	7
0	Del	3	5	7	9	↓	←	1
1	Ret	W	R	Y	I	P	*	Home
2	L	A	D	G	J	L	;	Ctrl
3	F8	4	6	8	0	↑	→	2
4	F1	Z	C	B	M	.	ESC	Space
5	F2	S	F	H	K	:	=	C=
6	F3	E	T	U	O	–	+	Q
7	@	Shft	X	V	N	,	/	Stop

Normale Tastaturmatrix bei C116, C16, Plus/4

BETRIEBSSYSTEM-ÄNDERUNGEN

Die Betriebssystem-Änderungen wurden so vorgenommen, daß ein Höchstmaß an Kompatibilität erreicht wird: Sogar der SCNKEY-Vektor sowie die Dekoder-Abfrage bleiben an der gleichen Adresse. Zusätzlich wurde noch ein deutscher Zeichensatz implementiert, die etwas unglückliche Belegung des Superkeyboards wurde an die des C16 bzw. des C64 angeglichen; am Zehner-Tastaturfeld steht neben der „Taschenrechner"-Belegung über die SHIFT-Lock-Taste noch eine Hex-Tastatur zur Verfügung. Die Grafikzeichenbelegung entspricht ebenfalls der des C16/C64.
Die TAB-Taste wird zur CTRL-Taste, die CTRL-Taste wird zur COMMODORE-Taste.
Wenn Ihnen die Belegung nicht gefällt, können Sie die Dekodiertabellen natürlich nach Belieben verändern. Einer Zeile im Listing entsprechen zwei Matrixzeilen, von oben nach unten gelesen.
Da jeder der vier neuen Tastaturdekodier-Tabellen 32 Bytes länger als die alte ist, und außerdem die SCNKEY- und DECODE-Routinen etwas umfangreicher wurden, mußte nach freien Stellen im ROM gesucht werden. Es gelang mir jedoch, die neuen Routinen so unterzubringen, daß keine andere Betriebssystem-Routine beeinträchtigt wird (auch die RS-232-Software bleibt voll erhalten!). Terry Ryan, Fred Bowen & Co. mögen es mir verzeihen, daß der Dekodiertabelle Nr. 3 auch ihre für das Betriebssystem

8255-Port				B0	B1	B2	B3	B4	B5	B6	B7	A0	A1	A2	A3	A4	A5	A6	A7
	Steckerpin auf Platine	U–Us.p. C=Centr.		U2	U3	U4	U5	U6	U7	U8	U9	C3	C4	C5	C6	C13	C12	C11	C10
		Verbindung	Diode	▶|	▶|	▶|	▶|	▶|	▶|	▶|	▶|	▶|	▶|	▶|	▶|	▶|	▶|	▶|	▶|
		Pull up 10 kΩ	Tastaturstecker	14	13	12	11	10	9	8	7	23*	21	20	19	18	17	16	15
C0	U12	⊕□—•	1	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	↓	↑	Home	RVS	Norm	Stop
C1	U13	⊕□—•	2	ESC	1	2	3	4	5	7	8	9	0	=	←	?	CE	*	/
C2	U14	⊕□—•	3	Tab	Q	W	E	R	6	U	I	O	–	←	→	7	8	9	–
C3	U15	⊕□—•	4		A	S	D	T	Y	J	K	L	P	]	DEL	4	5	6	+
C4	U16	⊕□—•	5	Shift	Z	X	F	G	H	M	,	;	[	Return	C=	1	2	3	Enter
C5	U17	⊕□—•	6	Ctrl		C	V	B	N	Space	.	?	,	π		0	.	00	
C6	U18	⊕□—•																	
C7	U19	⊕□—•																	

*:22 ≙ Kerbe im Tastaturstecker

Matrix der CBM600/700-Tastatur und Anschluß an User-Port-Platine (aus CW 10/86)

Für alle Interessenten: Das Systemlisting können Sie bei der Redaktion 16er, Postfach 1107, 8044 Lohhof, gegen Einsendung eines adressierten Freiumschlages (50 Pf. Porto) erhalten.

unrelevante Selbstverewigungs-Routine ab SCD89 zum Opfer fiel . . .

DIE HARDWARE

Grundlage der Schaltung ist die User-Port-/Centronics-Karte aus COMMODORE WELT Nr. 10 (Materialkosten: ca. 15 DM – den Centronics-Stecker brauchen Sie natürlich nicht!). Desweiteren benötigen Sie die wirklich sensationell preiswerte Tastatur der Commodore-Schrott-Rechner CBM 600/700 (bei Fa. Völkner, Postfach 5320, 3300 Braunschweig; Bestell-Nummer 9910965), außerdem 16 Dioden, 8 10K-Ohm-Widerstände, evtl. zwei Pfostensteckverbinder (im Artikel beschrieben), ein 16- und ein 6adriges Kabel sowie ein EPROM des Typs 27128, in dem das neue Betriebssystem untergebracht wird.

Aufbau der Hardware:

1. Bauen Sie die Userport-Platine wie in o.g. Artikel beschrieben auf – wenn Sie das geschafft haben, haben Sie schon 80% der Arbeit hinter sich!
2. Löten Sie – am besten auf der Platinen-Unterseite die Pull-Up-Widerstände ein.
3. Löten Sie (bei Bedarf) die Pfostensteckverbindungen an Platine und Kabel.
4. Löten Sie die 16 Dioden in die Kabelverbindung (z.B. zwischen Tastaturstecker und Kabel zur Platine).

EPROM BRENNEN

Nun müssen Sie nur noch das Betriebssystem-EPROM brennen (da sollte Ihnen ein Freund mit C-64 und Eprommer helfen können . . .): Wenn Sie keinen Assembler zum Ändern der Betriebssystemsoftware haben, gehen Sie bitte folgendermaßen vor (zum Austesten ist RAMOS, das RAM-Betriebssystem aus dem COMMODORE WELT-Sonderheft für C16/116/Plus/4, Juli 1986 sehr hilfreich!):

1. Springen Sie in den Monitor und geben Sie T C000 FFFF 2000 ein (dies gilt nur für 32- oder 64K-RAM. Bei der 16K Grundversion müssen Sie in 2 Schritten vorgehen: erst T C000 DFF 2000 und weiter mit Punkt 2. bis Adresse DFFF, dann abspeichern bis 4000, dann T E000 FFFF 2000 und weiter mit 2. ab Adresse E000, dann wieder – mit anderem Namen – abspeichern bis 4000).
2. Geben Sie nun mit dem M-Befehl die im Listing aufgeführten Änderungen ein.
   Beachten Sie, daß die Adressen der linken Spalte um hexadezimal $A000 vermindert eingegeben werden müssen, also statt CD89 z.B. 2C89, statt D4D8 34D8 usw. Außerdem sind die vier Dekodiertabellen ins Hexadezimalsystem umzurechnen (oder mit Basic-Programm direkt POKEn). Ansonsten geben Sie am besten nur die in der 2., 3. und 4. Spalte stehenden Hex-Zahlen nacheinander ein. Wenn Sie alle Änderungen eingegeben haben, speichern Sie die Daten für das EPROM mit
   S "TAST-EPROM",8,2000, 6000 ab.

Nun lassen Sie das EPROM mit diesen Daten brennen und ersetzen das alte Kernal ROM (beim C16/116: "U4", beim Plus4: "U24").
Wenn alles geklappt hat, werden auch Sie mit dem Super-Keyboard problemlos eingeben können:

```
10 PRINT "ES FUNKTIONIERT!"
20 GOTO  10
RUN
ES FUNKTIONIERT!
ES FUNKTIONIERT!
ES FUNKTIONIERT!
...
```

Ulrich Schäfer

Ä/Ö/Ü: FÜR IHREN C16/P4 KEIN PROBLEM

Die Tastatur des C16 oder Plus4 regt viele an, diesen Computer auch zum Schreiben von Briefen einzusetzen. Ganz besonders trifft dies auf den Plus4 zu, da in diesen bereits eine Textverarbeitung fest eingebaut ist.

Doch es gibt Probleme, nämlich mit den deutschen Umlauten. Auch unser Programm 'deutscher Zeichensatz für Script-Plus' aus Heft 6/87 konnte das Problem noch nicht komplett lösen, zum einen, weil nicht jeder sich Script-Plus zulegen will, zum anderen, weil nicht jeder einen Commodoredrucker besitzt und andere Drukker die deutschen Zeichen anders handhaben.

Wir können nicht für alle möglichen Kombinationen von Software und Druckern Lösungen anbieten, aber wir können Ihnen zeigen, wie eine Bildschirm-, Drucker- und Tastenanpassung funktioniert.
Um Ihnen die Sache zu erleichtern, haben wir die erforderlichen Routinen in einem Basicprogramm zusammengefaßt und dieses mit einem Anpassungsteil versehen. So können Sie eine Anpassung vornehmen, die ganz auf die von Ihnen benützte Software und Ihren Drucker zugeschnitten ist.

BILDSCHIRMANPASSUNG

Wie die Zeichen aussehen sollen, die auf den Bildschirm kommen, ist normalerweise im ROM im Adreßbereich von $D000-$D7FF (53248-55295) festgelegt. Schade, könnte man meinen, denn im ROM läßt sich leider nichts verändern. Jedoch kann man dem Computer auch sagen, daß er seinen Zeichensatz auch woanders, nämlich im RAM an einer anderen Stelle, suchen soll. Man braucht also nur den Zeichensatz aus dem ROM ins RAM kopieren und bestimmte Zeiger auf die neue Zeichensatzadresse umstellen.

Um zu wissen welche, brauchen wir ein paar zusätzliche Informationen, und zwar über den TED-Chip. Dieser Baustein ist unter anderem für die Bildschirmausgabe verantwortlich. Zwei von seinen Registern müssen wir manipulieren, nämlich Register $12 und $13. Da die Register des TED-Chips ganz einfach über bestimmte Speicheradressen angesprochen werden können, nämlich $FF00-$FF3F (65280-65343), wären dies die Adressen $FF12 und $FF13 (65298 und 65299).

TED-REGISTER $12 ($FF12 BZW. 65298)

Dieses Register enthält neben den oberen 5 Bits der Basisadresse der hochauflösenden Graphic (Bit 3-7) und neben den 2 oberen Bits für die Tonhöhe des Tongenerators 1 (Bit 0-1) das Bit Nr. 2, welches unsere besondere Beachtung verdient. Dieses Bit sagt nämlich dem TED-Chip, ob dieser den Zeichensatz im RAM oder im ROM suchen soll. Ist das Bit gesetzt, sucht er im ROM, ist es gelöscht, im RAM. Wir löschen es mit 'AND =$FB' (AND #%11111011), was wir in Basic durch 'poke 65298, peek (65298) and 251' realisieren. Damit wäre diese Sache schon geritzt. Ein kleiner Haken allerdings existiert, der weniger in fertigen, als hauptsächlich in fehlerhaften Programmen bzw. fehlerhaften Eingaben im Direktmodus zum Tragen kommt. Bei der Fehlerbehandlung wird die hochauflösende Graphic ausgeschaltet. Dabei wird leider auch wieder das Bit 2 auf 1 und somit auf ROM gestellt. Wenn unser neuer Zeichensatz einen anderen Adreßbereich belegt als der ROM-Zeichensatz, so kann man daraufhin nichts Sinnvolles mehr auf dem Bildschirm entdecken. Wollen wir dieses vermeiden, so genügt keine einmalige Umschaltung. Bei jeder Zeichenausgabe sollten wir diese vornehmen, in Form einer kurzen Maschinenroutine, oder noch besser, bei jedem Systeminterrupt.

TED-REGISTER $13 ($FF13 BZW. 65299)

Im TED-Register $13 sind die oberen 6 Bits der Zeichensatzbasisadresse zu finden (Bit 2-7). Auf die unteren 10 Bits müssen wir leider verzichten. 2 hoch 10 sind 1024 bzw. 1 Kilobyte. Das bedeutet, wir können unseren Zeichensatz nicht völlig beliebig, sondern nur in Schritten von 1 Kilobyte im Hauptspeicher positionieren. Hierzu ein Beispiel. Angenommen, unser Zeichensatz liegt ab 14 Kilobyte ($3800 bzw. 14336) im RAM. Wir müßten nun in die Adresse $FF13 den Wert =$38 bringen, also 'poke 65299,56'. Wollen wir jedoch Bit 0 und 1 nicht beeinflussen, so ist ein wenig mehr Aufwand zu treiben. Zuerst gilt es, die unteren 2 Bits mit 'and 3' zu isolieren, dann können wir mit 'or 56' (or #%00111000) die restlichen 6 Bits setzen. Angenommen, der zu setzende Wert sei 'X' un 'AD' sei 65299. Mit 'poke ad, (peek (ad) and 3) or x' wäre unser Vorhaben verwirklicht.

ZEICHENSATZ KOPIEREN

Wir haben hierzu verschiedene Möglichkeiten. Wir können es mit dem eingebauten Monitor tun, mit einem Basicpro- 

gramm, oder mit einem Maschinenprogramm.

Monitor:
'T D000, D7FF, 3800'

Basic:

```
10 poke 1177,62: rem le-
sen aus rom
20 for i=0 to 2047
30 poke 14336+i, peek
(53248+i): next
40 poke 1177,63:rem le-
sen aus ram
```

Da in Basic normalerweise der 'peek-Befehl' auf das RAM zugreift, ist zuerst eine Umstellung auf das ROM vonnöten. Hinterher empfiehlt sich wieder die Normaleinstellung.

Das Basicprogramm dauert etwas lang, das Monitorprogramm ebenfalls, sofern in den gebankten RAM-Bereich kopiert werden soll. Am schnellsten ist ein Maschinenprogramm, wie Sie es im Assemblerlisting 'Zeichensatz kopieren' vor sich haben.

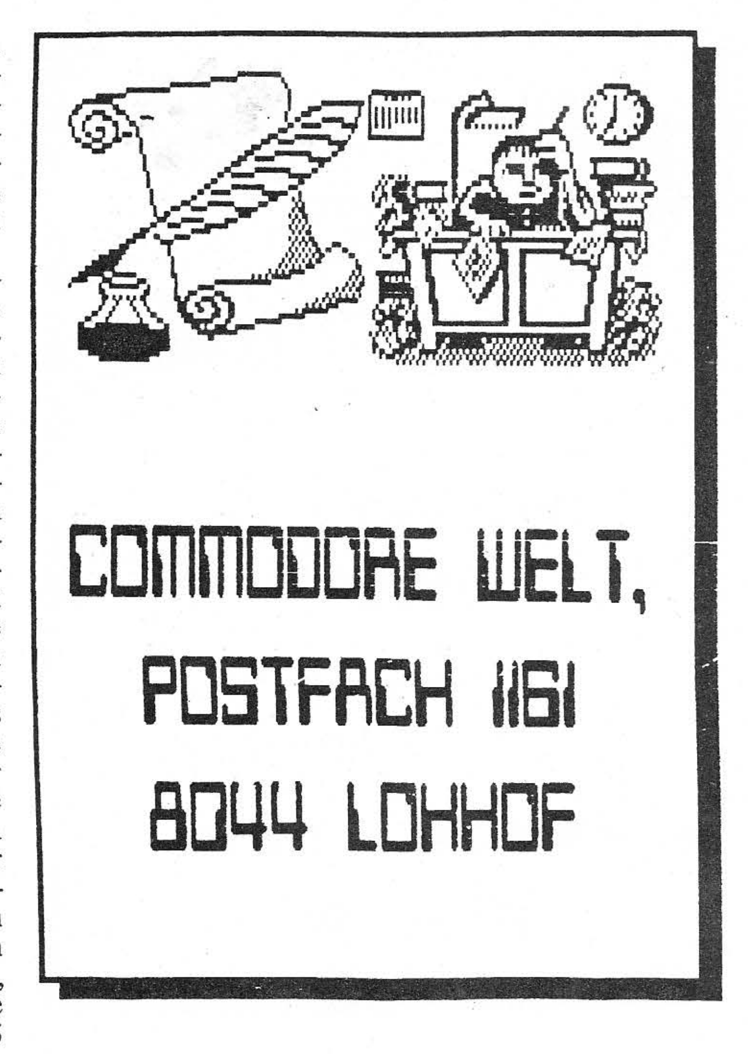
ZEICHEN ÄNDERN

Nachdem der Zeichensatz im RAM steht, ist es kein besonders Problem mehr, denselben auch zu ändern. Wir brauchen lediglich die richtigen Codes an die richtige Stelle zu schreiben. Hierzu betrachten wir uns einmal das Characterrom etwas näher. Wir können dieses mit dem Bitmonitor tun, den Sie ebenfalls in unserem Heft finden. Nehmen Sie dann außerdem Ihr Rechnerhandbuch zur Hand und schlagen Sie es bitte auf der Seite mit der Tabelle der Bildschirmcodes auf – bei mir ist dies die Seite 212 – kleine Abweichungen sind denkbar. Sie sehen dort zwei Zeichensätze, nämlich den Grafiksatz und den Klein-/Großbuchstabensatz mit jeweils 128 Zeichen. Wenn Sie nun diese 256 mit 8 multiplizieren, so kommen Sie genau auf 2 Kilobyte, den Speicherbedarf des Bildschirmzeichensatzes. Wie die Überprüfung mit dem Bitmonitor ergibt, sind die Zeichenmuster im Speicher in ebenderselben Reihenfolge abgelegt wie in unserer Tabelle, nämlich je Bytes Bitmuster für die 128 Zeichen von Satz 1 und dann dasselbe für Satz 2. Da wir in unserem Falle an Satz zwei interessiert sind, finden wir die Adresse des zu einem Buchstaben gehörenden Bitmusters dadurch, daß wir den Bildschirmcode mit 8 multiplizieren und den Wert 1024 als auch die Zeichensatzbasisadresse hinzuaddieren. Dieses Byte und die folgenden sieben sind nun durch die neuen Bitmuster zu ersetzen. Das Nachsehen in der Tabelle der Bildschirmcodes kann man sich durch einen kleinen Trick sparen. Wenn Sie einen Blick auf die Tabelle mit den ASC- und CHR$-Codes werfen, so sieht es zwar nicht so einfach aus, daraus den Bildschirmcode zu bekommen. Wenn Sie aber mit "SCNCLR' den Cursor in die obere linke Bildschirmecke befördern und dort ein Zeichen auf den Bildschirm schreiben, z.B. mit print 'a', so steht nun im Bildschirmspeicher an erster Stelle, also an Adresse $-C00 (3072) der Bildschirmcode, den wir zur Berechnung der Zeichenmatrixadresse brauchen. Dorthin 'poken' Sie die entsprechenden Zeichencodes, welche Sie von uns übernehmen können, die Sie aber auch mit unserem Programm 'Bitmonitor' selbst bestimmen können.

DRUCKERANPASSUNG

Damit ein Drucker deutsche Umlaute drukken kann, braucht er einen deutschen Zeichensatz. Die meisten Drucker bieten mehrere Zeichensätze zur Auswahl, welche meist mit DIP-Schaltern vorgenommen wird. Nähere Einzelheiten hierüber stehen in Ihrem Druckerhandbuch. Was auch noch in Ihrem Druckerhandbuch steht, sind die Codes, die Ihr Drucker braucht, um gewisse Zeichen – in unserem Falle die deutschen Zeichen – zu drucken. Manchmal lassen sich diese Codes bereits über die Tastatur eingeben, oft aber auch nicht, da es 256 Codes gibt, Tasten aber weniger, oft weigert sich auch Ihre Textverarbeitung, gewisse Codes zu akzeptieren. Eine Lösung bietet die Druckeranpassung. Wir nehmen eine beliebige Taste für unser zu druckendes Zeichen her. Wenn das Zeichen für die Druckerausgabe bereitsteht, machen wir sozusagen dem Drukker ein X für ein U vor, indem wir gegebenenfalls dieses Zeichen durch ein anderes ersetzen. Dies ist hier nun nur durch ein Maschinenprogramm möglich. Unter gewissen Voraussetzungen können wir uns in die Ausgaberoutine einschalten, und zwar dann, wenn ein Maschinensprachprogram-

mierer, was er am besten auch tun sollte, die Kernalroutinen benutzt, anstatt Betriebssystemroutinen direkt anzuspringen. Das Kernal ist eine Sprungliste, die allen Commodorecomputern, Amiga und PC mal ausgenommen, gemeinsam ist. Ob nun eine Zeichenausgabe auf einem VC20 oder einem Plus4 erfolgen soll, es geht haargenauso. Diese Routine heißt 'BSOUT' (Basic Output), oder auch 'CHROUT' (Character-Out), zu deutsch Zeichenausgabe. Mit 'JSR $FFD2' wird die Ausgabe des im Akku befindlichen Zeichens auf das Ausgabegerät veranlaßt. Das Schöne ist nun, daß die irgendwo im Speicher plazierte Routine ($EC4B beim C16) nun nicht direkt von der Sprungliste aus angesprungen wird, sondern indirekt über einen Sprungvektor 'IBSOUT', welcher beim C16 die Adressen $324-$325 (804-805) belegt. Wir brauchen dort nur anstelle der Routinenadresse $EC4B eine andere vermerken, z.B. $0138, und schon springt das Programm in unsere Routine, die, bevor sie im Falle einer Druckerausgabe das Zeichen ausgibt, anhand einer Liste überprüft, ob dieses nicht besonders zu behandeln sei. In diesem Falle wird es durch ein entsprechendes Zeichen aus einer zweiten Liste ersetzt. Weitere Einzelheiten können Sie der als Assemblerlisting vorliegenden Druckerroutine entnehmen.



TASTATURANPASSUNG

Wir dachten, mit der Druckeranpassung das Problem gelöst zu haben. Mit der eingebauten Software erschienen wunderbar unsere Zeichen auf dem Bildschirm, auf dem Drucker suchten wir sie vergebens. Die Programmierer der Software hatten leider nicht die Kernalroutinen benutzt, weshalb das Programm unsere Routine verschmähte. Doch auch hier gab es eine Pforte, durch die wir uns Zugang zum System verschaffen konnten, zwar nicht zur Ausgaberoutine, jedoch zur Tastaturabfrage. Wie Sie vielleicht wissen, erfolgt 50 oder 60 mal in der Sekunde ein sogenannter Systeminterrupt, im Zuge dessen nicht nur Rekorder, Uhr und Sound bedient werden, sondern auch die Routine SCNKEY aufgerufen wird, die dafür sorgt, daß, sofern eine Taste gedrückt ist, deren ASC-Code auch in den Tastaturpuffer wandert – die Funktionstasten wollen wir hierbei einmal unberücksichtigt lassen. Bevor nun vom Anwenderprogramm aus ein Zugriff

TASTATUR – KONTRA DRUCKERANPASSUNG

auf den Tastaturpuffer erfolgen kann, tauschen wir bereits gegebenenfalls den dort zuletzt abgelegten Code aus. Der Computer bekommt so bereits bei der Eingabe den Code, den der Drucker später bei der Ausgabe braucht.

Bei der Druckeranpassung kann es vorkommen, daß Sie gewisse Tasten nicht verwenden können, weil die Software gewisse Codes nicht akzeptiert. Bei der Tastaturanpassung können Sie alle Tasten benützen. Wenn aber die Software gerade die Codes verweigert, die ihr Drucker braucht, so haben Sie Pech gehabt. Die Lösung dieses Problems bietet eine kombinierte Tastatur- und Druckeranpassung. Da wir unseren Stack mit der Tastaturanpassung schon ziemlich ausgenützt haben, nahmen wir nun auch noch den Basic-Pseudo-Stack für die Druckeranpassung her. Ob das wohl gutgeht? Es funktioniert ganz hervorragend, wovon Sie sich leicht selbst überzeugen können. Wie die Anpassung zu handhaben ist, erfahren Sie in den 'Bedienungshinweisen für das Zeichensatzprogramm'.

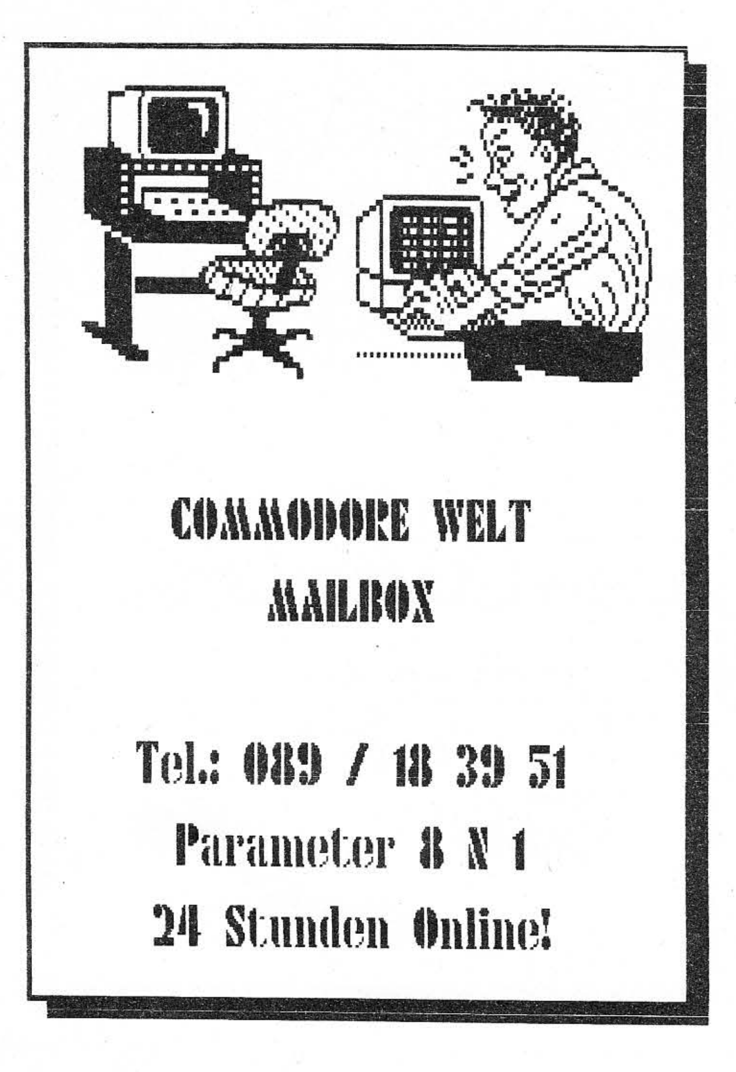
BEDIENUNGSHINWEISE FÜR ZEICHENSATZ-PROGRAMM

Wenn Sie einen Drukker mit deutschem Zeichensatz besitzen, ob dieser nun COMMODORE oder sonstwie heißt, mit diesem Programm bleiben keine Wünsche mehr offen. Vielfältige Anpassungsmöglichkeiten erlauben Ihnen nicht nur ein unbeschwertes Schreiben von deutschen Umlauten. Wenn Ihr Drucker weitere Zeichen kennt, so können Sie ihm auch diese abverlangen, ohne auf sonstige ASCII-zeichen wie "@ £ * ↑ π <>," verzichten zu müssen.

Wenn Sie allerdings zu denjenigen gehören, deren Drucker zwar keinen deutschen Zeichensatz besitzt, jedoch grafik-

fähig ist, so kann Ihnen zwar geholfen werden, jedoch wird dies noch einige Zeit dauern, bis ein Programm dafür geschrieben ist.

Seitdem ich meinen deutschen Zeichensatz habe, ist mir Script/Plus die liebste Textverarbeitung geworden, obwohl mir auf dem C128 noch weitere Textverarbeitungen zur Verfügung stehen. Die eine stürzt bisweilen ab, die andere nimmt Tastendrücke, welche zu schnell aufeinanderfolgen, nicht an. Die deutschen Umlaute bringe ich nun sogar mit der eingebauten Software des Plus4 auf Bildschirm und Drucker. Jedoch kann es hier sein, daß nicht jeder Drucker mitmacht, weil nicht jedes Zeichen von der Software angenommen wird.



ANPASSUNGSFÄHIG WIE EIN CHAMÄLEON

Das Zeichensatzprogramm hat, wie man sieht, gegenüber demjenigen in Heft 6/87, welches nur auf einem Commodoredrucker wie dem MPS 1000 funktionierte, um einiges in der Länge zugenommen. Es besteht aus den drei Abschnitten Anpassung, Routinensammlung und Installation. Besonders interessant ist der Abschnitt „Anpassung", weil Sie in diesem Ihre individuellen Anpassungen vornehmen können.

ZEICHENSATZ-ADRESSE VERSCHIEBBAR

Der Zeichensatz ist in Schritten von 1 Kilobyte verschiebbar und belegt einen Speicherbereich von 2 Kilobyte. Das erste Kilobyte beinhaltet den Grafikzeichensatz, das zweite den Kleinbuchstabenzeichensatz. Für Script/Plus haben wir in Zeile 135 die Zeichensatzadresse auf 14 Kilo-Byte ($3800/14336) gelegt. Wenn wir einen C16 mit Erweiterung oder einen Plus4 besitzen, können wir auch 30 Kilobyte ($7800) wählen. Im Bereich über $8000 liegt Script/Plus, so daß uns dieser nicht offen steht. Wenn wir mit einem anderen Programm zusammenarbeiten, müssen wir eventuell den Zeichensatz an eine andere Adresse legen, damit dieser nicht mit dem Programm oder sonstigen Daten kollidiert. Für die eingebaute Textverarbeitung des Plus4 eignet sich die Adresse 60 Kilobyte ($F000).

BASICENDE VERLEGEN

Wenn wir den Zeichensatz hinter ein Basicprogramm legen und den Basic-Bereich nicht einschränken, so ist es möglich, daß Basicvariablen plötzlich beginnen, unseren Zeichensatz zu überschreiben. Wenn wir dieses verhindern wollen, so brauchen wir nur das "n" in Zeile 155 durch ein "j" zu ersetzen.
Wenn es Platzprobleme gibt, so kann man den Grafikzeichensatz opfern und auch per Hand das Basicende festlegen, indem man die entsprechenden Werte in die Speicherstellen $35-$36 (53-54) schreibt.

TASTATUR-ANPASSUNG

Es gilt nun, dem Drukker zum richtigen Code zu verhelfen. Das Zeichen, welches der Computer per Tastendruck bekommt, gilt es entsprechend auszutauschen. Dieses kann gleich bei der Eingabe geschehen, indem wir das im Tastaturpuffer zuletzt abgelegte Zeichen flugs durch das richtige ersetzen, bevor der Computer überhaupt in der Lage ist, dieses dem Puffer zu entnehmen. Wir tun dies im Zuge des Systeminterrupts, so daß dieses Verfahren auch funktioniert, wenn der Programmierer der Software, für welche wir den geänderten Zeichensatz benützen wollen, einfach die Kernal-Routinen unterlaufen hat. Da Letzteres bei der eingebauten Software des Plus4 der Fall ist, so schreiben Sie bitte "t" in die Zeile 200, wenn Sie diese Software benützen wollen. Ob die eingebaute Software allerdings gerade die Zeichen auch annimmt, welche Ihr Drucker benötigt, um die deutschen Umlaute auch auszugeben, ist Glücksache. Der Commodore MPS 1000 machte jedenfalls mit. Wenn Ihr Drucker dieses nicht tut, und Sie sich auch keinen anderen anschaffen wollen, so wechseln Sie am Besten die Textverarbeitung. Bei einer ordentlich programmierten können Sie noch eine weitere Anpassungsart fahren, nämlich die Druckeranpassung.

Wählen Sie "d" in Zeile 200, so wird das einge-

gebene Zeichen so, wie es von der Tastatur kommt, vom Computer angenommen. Der Austausch erfolgt erst bei der Druckerausgabe. Jeder Drucker mit deutschem Zeichensatz und seriellem Anschluß ist nun verwendbar. Allerdings kann es sein, daß die verwendete Software nicht jede Taste annimmt. Wenn Sie mit jemandem, der dieselbe Textverarbeitung benutzt wie Sie, Text auf Diskette austauschen, so gibt

DRUCKERANPASSUNG

es Probleme mit den Umlauten, sofern der andere eine andere Tastenbelegung vorzieht. Jedoch gibt es noch einen Ausweg aus diesem Dilemma, nämlich die Kombination von Tastatur- und Druckeranpassung.

SPEZIALANPASSUNG

Mit der Tastaturanpassung können Sie dem Computer bei beliebigem Tastendruck jedes beliebige Zeichen unterjubeln. Sie brauchen nur dafür zu sorgen, daß der Computer ein Zeichen erhält, welches die Software annimmt. Am Besten ist ein geeigneter Grafik-Code, da man in diesem Falle kein Textzeichen zu opfern braucht. Wir haben in den Zeilen 360 - 365 bereits eine Anpassung vorgenommen, die bei Script-Plus funktioniert. Sie sollten diese, sofern Sie Script-Plus verwenden, nicht verändern, da, wenn jeder die gleichen Codes verwendet, auch ein Datenaustausch auf Diskette oder Kassette möglich ist, ohne daß Ihre Tastaturbelegung und Druckeranpassung hierbei eine Rolle spielen. Die Spezialanpassung ist die komfortabelste und setzt außerdem zumindest für Script/Plus einen Standard. Sie ist daher, sofern verwendbar, den anderen Anpassungen vorzuziehen.

DRUCKERCODES

Damit Ihr Drucker deutsche Umlaute überhaupt von sich geben kann, müssen Sie ihn auf deutschen Zeichensatz, meist mittels Dipschalter, umstellen. Wie das gemacht wird, steht in Ihrem Druckerhandbuch. Dort finden Sie des weiteren eine Tabelle, aus welcher hervorgeht, welche ASCII-Code der Drucker braucht, um die gewünschten Umlaute darzustellen. Wenn diese Werte von denen in den Zeilen 220 - 225 abweichen, so tragen Sie dort bitte die richtigen ein.

ZEICHENMUSTER

In den Zeilen 245-340 ist vermerkt, wie die Zeichen aussehen. Das Bitmuster können Sie errechnen, wenn Sie die angegebenen Dezimalzahlen in Dualzahlen umrechnen. Sie brauchen keine Änderung der Zeichenmuster vornehmen.

SPEZIALCODES

Die Spezialcodes spielen nur eine Rolle für die Spezialanpassung, sie sind auf Script/plus abgestimmt. Ändern Sie sie bitte nicht für Script/Plus, damit wir einen Standard haben. Falls für eine andere Soft-

Zeichensatz C 16

```
50 rem zeichensatz========c16=  <ni>
55 rem (p) 6/87 commodore welt  <bm>
60 rem ======================== <eg>
65 rem     by a.mittelmeyer ==  <kf>
70 rem                      ==  <ep>
75 rem                      ==  <fe>
80 rem basic v3.5           ==  <kh>
85 rem c16/116/plus4        ==  <hg>
90 rem ======================== <ge>
95 :                            <jj>
100 rem *********************** <ip>
105 rem    anpassung            <ki>
110 rem *********************** <jj>
115 rem ======================= <ea>
120 rem neue zeichensatz-       <kj>
125 rem adresse in kilobyte     <an>
130 rem ----------------------- <op>
135 data 14                     <gp>
140 rem ======================= <fj>
145 rem basicende verlegen j/n  <ln>
150 rem ----------------------- <ad>
155 data n                      <gm>
160 rem ======================= <gn>
165 rem   tastenwahl            <cp>
170 rem ----------------------- <bh>
175 data "[\]^<@>$%&"           <bb>
180 rem ======================= <ib>
185 rem tasten-, drucker-,      <nj>
190 rem spezialanpassung t/d/s  <fo>
195 rem ----------------------- <da>
200 data s                      <jo>
205 rem ======================= <jk>
210 rem   druckercodes          <np>
215 rem ----------------------- <ee>
220 data 251,252,253,254,219    <bb>
225 data 220,221,240,237,222    <af>
230 rem ======================= <ld>
235 rem   zeichenmuster         <ga>
240 rem ----------------------- <fn>
245 data 0,102,60,6             <fl>
250 data 62,102,62,0:   rem "ae <af>
255 data 0,102,60,102           <mc>
260 data 102,102,60,0:  rem "oe <eh>
265 data 0,102,0,102            <jh>
270 data 102,102,62,0:  rem "ue <fj>
275 data 0,60,102,124           <nl>
280 data102,102,124,96:rem "ss  <nn>
285 data 102,24,60,102          <bh>
290 data 126,102,102,0:rem "Ae  <ak>
295 data 102,60,102,102         <eo>
300 data 102,102,60,0:  rem "Oe <op>
305 data 102,0,102,102          <cc>
310 data 102,102,60,0:  rem "Ue <pp>
315 data 60,102,110,110         <ga>
320 data 96,98,60,0   : rem "@  <oj>
```

ware doch ein Umstellen erforderlich sein sollte, so brauchen Sie nur einen x-beliebigen Code hier einzusetzen, der von der Software angenommen wird.

WAS SIE ANPASSEN MÜSSEN

Bei Script-Plus nur die Druckercodes. Bei der eingebauten Software des Plus4 zusätzlich die Zeichensatzadresse und die Anpassungsart.

WAS SIE ANPASSEN SOLLTEN

Empfehlenswert ist eine andere Tastenbelegung, ich z.B. nehme beim Plus4 die Tasten " @ Ł*" für die Umlaute her, da diese so schön beieinanderliegen. In Verbindung mit der Shifttaste bekomme ich die großen Umlaute.

WAS SIE ANPASSEN KÖNNEN

Sie sind durchaus nicht auf deutsche Umlaute beschränkt. Sie können,

```
905 for i=1822 to 1821+n              <dn>
910 reada:pokei,a:next                <jm>
915 poke1814,29:poke1815,7            <jf>
920 poke1802,n:poke1804,111+n         <mm>
925 poke 804,0:poke 805,7             <aa>
930 rem ======================        <hc>
935 rem basicende                     <lh>
940 rem ----------------------        <bm>
945 restore155:readx$                 <bi>
950 ifx$="n" then end                 <cl>
955 poke 54,x:poke 56,x               <ac>
960 poke 53,0:poke 55,0               <lf>
965 rem ======================        <jf>
970 rem 012277 bytes memory           <op>
975 rem 004465 bytes program          <dd>
980 rem 000070 bytes variables        <kn>
985 rem 000165 bytes strings          <ei>
990 rem 007577 bytes fre(0)           <ip>
995 rem ======================        <ld>
```

man staune, sogar noch ein paar Zeichen hinzunehmen. Sie können natürlich auch welche wegnehmen. Wenn Sie allerdings zuviele neue Zeichen hinzufügen, kann es Kollisionen mit dem Programm geben. Mein Programm ist nicht dazu gedacht, eine vollständige Umcodierung der Tastatur vorzunehmen, sondern beschränkt sich auf das Nötige. Um es kurz und einfach zu halten, wurde auf eine Anpassung mit Menüführung verzichtet. Daher ist vielleicht das Anfügen eines neuen Zeichens auch etwa umständlich. Auch wenn Sie nur eine Taste umbelegen wollen,



ist es nötig, auch den Druckercode anzupassen, welcher im Normalfall mit dem ASCII-Code des Commodore Computers identisch sein sollte.

ANPASSUNGSVORSCHLÄGE

Plus4
ä : (@)
ö : (£)
ü : (*)
ß : (1)
Ä : (shift)
Ö : (shift (£)
Ü : (shift) (*)
@ : (cbm) (@)
£ : (cbm) (£)

* : (cbm) (*)
1 : (cbm) (+)

C16
ä : (@)
ö : (+)
ü : (–)
ß : (1)
Ä : (shift)
Ö : (shift) (+)
Ü : (shift) (–)
@ : (cbm) (@)

\+ : (cbm) (+)
– : (cbm) (–)
↑ : (cbm) (P)

Mit (cbm) ist hierbei die Commodore-Taste gemeint.

WEITERE ZEICHENMUSTER

Für die Anpassung sind noch zusätzliche Zeichenmuster erforderlich.

Plus4:
341 data 0,102,60,255,60,102,0,0: rem*
342 data 0,24,60,126,24,24,24,24:rem ↑

C16:
341 data 0,24,24,126,24,24,0,0: rem +
342 data 0,0,0,126,0,0,0,0: rem –
343 data 0,24,60,126,24,24,24,24:rem ↑

DRUCKER- UND SPEZIALCODE-ERGÄNZUNG

Plus4:
226 data 42,94
366 data 42,94

C16:
226 data 43,45,94
366 data 43,45,94

EIGENE ZEICHEN ENTWICKELN

Eigene Zeichen entwickeln und ausdrucken können Sie mit diesem Programm noch nicht. Bis jetzt sind Sie noch auf den Druckerzeichensatz angewiesen. Ab nächstem Sonderheft werden wir diese Beschränkung durchbrechen und Sie können, wenn Sie wollen, auch kyrillisch, chinesisch oder in Runen schreiben und drucken.

A. Mittelmeyer

COMMODORE FLOPPY 1551: SCHNELLERE DATEN FÜR DIE 16er UND PLUS 4

Mit modernerer Technik wartet die Floppy 1551 auf

Schon auf der CES im Frühjahr 1984 in Las Vegas tauchte diese Floppy, unter der Bezeichnung SFS 481, als Baustein einer neuen Computerreihe auf. Und zwar als Diskettenlaufwerk zum damals geplanten Topmodell CV 364, aus dem sich wahrscheinlich der Plus 4 entwickelte. Bilder dazu brachte eine Computer-Zeitschrift schon in ihrem 84'er Frühjahrsheft. Die Floppy ist, um es kurz zu sagen, nicht für den Anschluß an den VC-20, C-64 oder PC-128 gedacht, sondern für die kleineren Brüder, den C-16, C-116 und den Plus 4. Wofür brauchen die denn eine eigene Floppy? Die 1541 ist problemlos auch an diese Computer anzuschließen. Obwohl diese Floppy genauso aussieht wie die 1541, außer der Farbe, die den Computerfarben entspricht, weist sie doch eine modernere Technik auf. Als IEC-Bus Variante wird die Floppy, mit einem leicht geändertem DOS, nämlich am Expansionsport der Computer C-16, C-116 und Plus 4 angeschlossen. Technikfreaks werden jetzt hellhörig, denn am Expansionsport liegt ja der Adress- und Datenbus des Prozessors an. Die Daten zur Floppy werden nämlich nicht mehr seriell, sondern parallel übertragen, was eine höhere Übertragungsrate bewirkt. Die Übertragungsrate beträgt jetzt 1600 Bytes/Sec. (Im Gegensatz zu den doch mageren 300 - 400 Bytes /Sec. der 1541). Im Klartext bedeutet das, die 1551 ist jetzt ca. 3 mal so schnell wie die doch behäbig wirkende 1541 in Normal-Version. Warum diese Floppy nur für den C-16 usw. zu erhalten ist, erklärt sich mit der in England größeren Verbreitung dieser Computer, vor allem des Plus 4, der dort zu Höhenflügen ansetzt. Um dem Rechnung zu tragen, mußte Commodore wahrscheinlich zu dieser Maßnahme greifen und ein Laufwerk speziell für den Plus 4 anbieten, damit man das Betriebssystem des Plus 4, das ja voll auf die Diskette ausgerichtet ist, auch in gebührendem Maße ausnutzen kann. Obwohl der Plus 4 einen Userport sein eigen nennt, hat man vielleicht mit Rücksicht auf die Computer C-16 und C-116, die intern ja fast identisch mit dem Plus 4 sind, den Expansionsport gewählt. Warum die Floppy 1551 gerade jetzt auf dem deutschen Markt erscheint, obwohl es sie in England schon seit 1984 gibt, kann man nur mutmaßen. Vielleicht hängt es mit der mittlerweile großen Verbreitung der C-16 Reihe zusammen, von denen es ja mittlerweile ca. 250 000 Stück auf dem deutschen Markt gibt.



Die neue 1551

Da die Floppy schreib- und lesekompatibel zur 1541 ist, können alle Disketten, die mit dieser beschrieben wurden, auch auf der 1551 gelesen werden. Umgedreht geht es genauso. Da ja der serielle Port nicht von der Floppy belegt ist, kann hier auch noch eine 1541 angeschlossen werden, die dann aber eine andere Adresse bekommen muß, da die Adresse 8 auch weiterhin für die 1551 zur Verfügung steht. Im Zusammenhang mit der parallelen Übertragung und der moderneren Technik, die sich in dem Prozessor 6510T und dem Portbaustein 6523A niederschlägt, ergibt sich eine Formatierungszeit von ca. 23 Sec. Da geht gegenüber der normalen Floppy 1541 mit ca. 83 Sec. die Post ab. Die 15 51 wird mit Hilfe eines Interface, das fest über ein Kabel mit der Floppy verbunden ist, an den Computer angeschlossen. Da der Expansionsport im Interface durchgeschleift ist, können hier immer noch Module oder andere 1551 angeschlossen werden.

SCHALTEN MIT DEM C16

Wie Sie mit einfachen Mitteln auch komplizierte Steuerungsvorgänge realisieren können, ist Inhalt dieses Berichtes. Der Autor arbeitet privat auf den Rechnern Plus 4 und C 16 und verwendet diese für die diversesten Steuerungsaufgaben. Sein neuester Streich: Kamerasteuerung via Kassettenport!

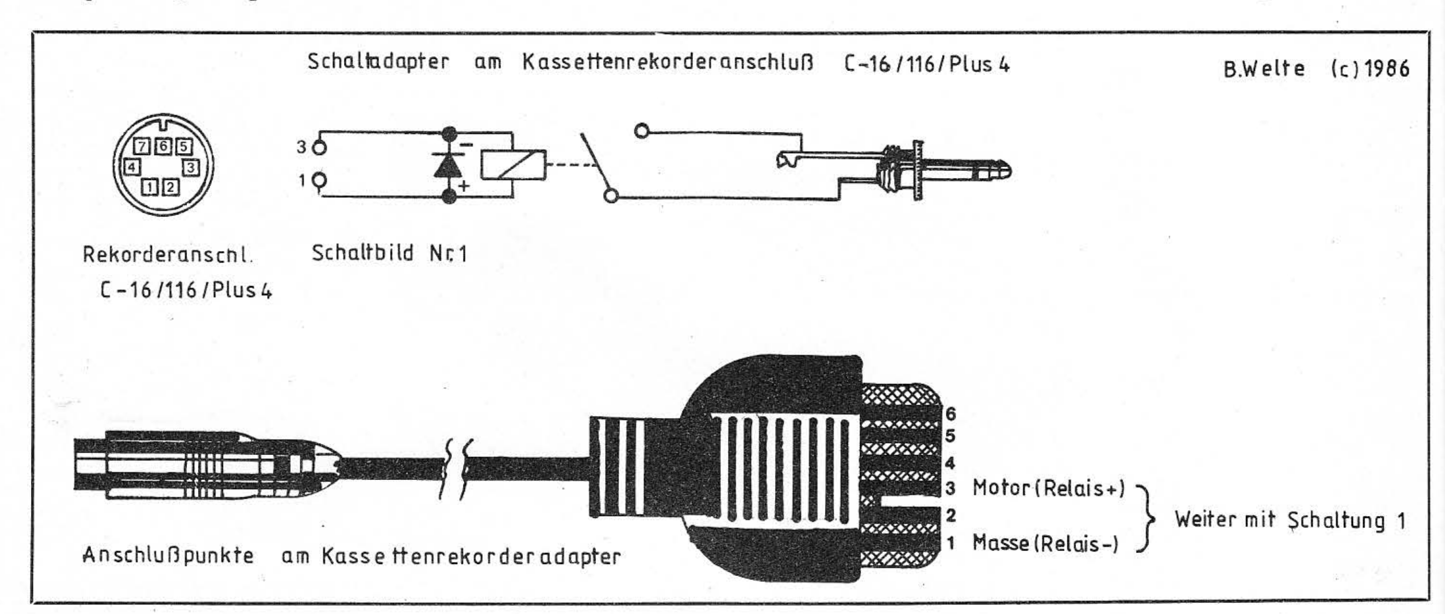
Da die Firma Commodore ihre Computertypen der Reihe C16/116 nicht mit einem Userport ausgestattet hat, ergeben sich mit diesen Geräten immer einige Schwierigkeiten, einen Schaltvorgang, wie auch immer, an die Außenwelt weiter zu geben. Einige findige Köpfe haben sich dieselben zerbrochen und bieten einige Lösungen an, die am Expansionsport betrieben werden. Siehe auch Heft 10/11 der COMMODORE WELT. Allerdings bedingen diese Möglichkeiten immer einen Eingriff in den Computer, um die eine oder andere Verbindung herzustellen oder aufzutrennen. Alles dieses behagte mir nicht besonders und so suchte ich nach einer anderen Lösung für mein Problem.

PROBLEMLÖSUNG GESUCHT UND GEFUNDEN

Als Schmalfilmamateur suchte ich eine Möglichkeit, bei Trickaufnahmen die Übersicht über die gemachte Bilderzahl, Filmlaufzeit und Filmlänge zu behalten. Auch sollte der Computer die Steuerung bzw. die Auslösung der Filmkamera übernehmen. Nach einigem Nachdenken kam mir die Idee, den Kassettenport zu diesem Zweck heranzuziehen. Nach einem Blick in den Schaltplan fand ich dann die Lösung. Mir fiel nämlich auf, daß der Computer den Kassettenmotor über einen Leistungstransistor ein- und ausschaltet. Von den Daten her mußte dieser in der Lage sein, auch ein Relais zu schalten. Über den Relaiskontakt könnte dann die Filmkamera gesteuert werden. Gesagt, getan. Ein Adapter, mit einem kleinen Relais, war schnell gebastelt und wurde einem Test unterworfen, der zur vollen Zufriedenheit ausfiel. Die Probleme begannen aber erst, als die Schaltung am Computer getestet werden sollte. Es fehlte nämlich die Adresse, mit der der Ausgang angesteuert werden konnte. Die vorhandenen SAVE- und LOAD-Routinen waren nicht geeignet (Abschaltung des Bildschirms, Zerstörung des Bildschirmaufbaus) und so mußte nach einem anderen Weg gesucht werden. Den fand ich dann in der Adresse 64784. Wenn man also diese Adresse anspricht und zwar mit dem Befehl >Poke 64784,2<, wird der Ausgang eingeschaltet und mit >Poke 64784, 255< wieder ausgeschaltet. Schaltung (1) zeigt den einfachen Aufbau des kleinen Interfaces zum Anschluß der Filmkamera an den Computer. Er erfolgt mit einem 8poligen Stecker, wie er sich auch am Kassettenrekorder befindet. Bild (2) zeigt eine Lösung, die die etwas schwierige Beschaffung des Steckers erübrigt. Hierbei handelt

MINI-INTERFACE ZUM EIGENBAU

es sich um einen handelsüblichen Adapter, wie zum Anschluß der C-64-Datasette am C-16 vorgesehen ist. Hier ist es leicht möglich, an die entsprechenden Leitungen zu kommen. Ab hier geht es dann mit der Schaltung aus Bild (1) weiter.



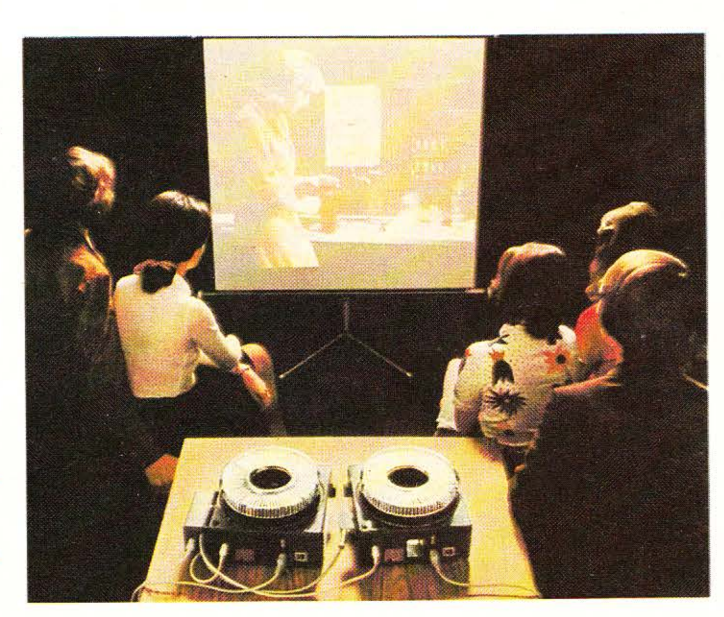
So einfach ist das Schaltbild des Interfaces für den Kassettenport. Mit diesem Interface wird es möglich, externe Geräte anzusteuern.

DIA SHOW PER COMPUTER

Strahlende Dias auf der Leinwand – das ist immer noch ein Höhepunkt für einen Hobby-Fotografen. Wer sich an den projizierten Dias wirklich und auf Dauer erfreuen will – vor allem bei Vorführungen mit größerem Zuschauerkreis, muß sich aber auch um gute Ergebnisse bemühen. Dazu gehört aber auch eine gute Vorführung, eventuell mit der entsprechenden Ton- und Geräuschkulisse. Die Tonbildschau wird zur echten „Show", wenn statt des monotonen Diawechsels die Projektion ohne Dunkelpause erfolgt. Gilt für das Bild-Pause-Bild-Staccato die Empfehlung, eine Vorführung auf 100 Dias zu limitieren, so dürfen es hier je nach Qualität auch 300 und mehr sein.
Jetzt entsteht der Eindruck des „Fließens" wie beim Film, und trotzdem ist das Erlebnis noch stärker, noch eindrucksvoller. Man sollte jedoch auch schon bei der Aufnahme

JETZT WIRD'S ELEKTRONISCH

darauf achten, daß später viele verbindende Dias verfügbar sind. Nicht das Einzelmotiv ist hier entscheidend, wie bei der Mono-Diaserie, sondern das harmonische Entstehen eines Bildes aus dem Vorausgegangenen.
Die Grundausstattung einer Überblendanlage setzt sich aus zwei vollautomatischen Diaprojektoren und dem sie verbindenden Steuergerät zusammen. Im Prinzip lassen sich alle Projektoren, die mit 24 Volt Gleichspannung betrieben werden, verwenden. Entweder sind diese Geräte bereits für Überblendung eingerichtet, oder sie können problemlos umgerüstet werden. Um die Überblendtechnik haben sich viele Hobby-Fotografen Gedanken gemacht und sind zu mehr oder weniger guten Ergebnissen gekommen. Inzwischen gibt es von den verschiedensten Herstellern Überblendgeräte, die sich aber teilweise durch einen so hohen Preis auszeichnen, daß manche Hobby-Fotografen abgeschreckt werden und nach einer anderen Lösung suchten und sie auch gefunden haben. Dabei ist der Aufwand nicht einmal so groß. Mit ein paar elektronischen Bauteilen und etwas Geschick läßt sich eine Schaltung aufbauen, die am Userport eines Computers betrieben werden kann. In diesem Falle ist es der Plus/4. Aber auch jeder andere 8-Bit-Computer, der eine vergleichbare Schnittstelle besitzt, ist hierfür geeignet.



Fotoamateure wissen es: Überblendprojektoren für Diashows sind teuer. Es geht auch preiswerter, nämlich mit Ihrem Plus4, wenn Sie über einige Lötkolbenkenntnisse und Elektronik-Wissen verfügen. Unser Bericht sagt Ihnen, wie Sie sich eine derartige Schaltung selbst bauen können.

Erschrecken Sie bitte nicht, wenn Sie die Schaltung (Bild 1) betrachten. Es umfaßt die komplette Schaltung, die sich auf einer Lochrasterplatine oder mittels der Fädeltechnik aufbauen läßt.
Die Schaltung benötigt eine eigene Stromversorgung, da sonst die Phasenschnittsteuerung für die beiden Projektorlampen nicht verwirklicht werden kann. Jeder der beiden Projektoren muß so angeschlossen werden, daß der Dia-Vorwärtstransport durch die Kontakte Change 1 und 2 möglich ist (Bild 2). Sollten die Projektoren nicht für eine Überblendung eingerichtet sein, kann dieses durch das Einbauen der beiden TRIAC's in die Lampenstromkreise nachgeholt werden. Hierbei ist auf eine gute Wärmeableitung zu achten, denn durch eine solche Projektorlampe fließen nämlich bei 24 V und 150 W satte 6,2 A. Beim Einschalten aber das bis zu 15fache, also nahezu 100 A.

WIE FUNKTIONIERT'S NUN?

Die digitalen Informationen, die am Ausgang des Computer (Userport) anliegen, werden mit Hilfe von IC's in analoge Signale umgewandelt, abhängig von dem anliegendem Bitmuster, mit der dann der TRIAC angesteuert wird. Um eine Überblendung zu erreichen, wird jeweils ein TRIAC gesperrt und der andere durchgesteuert. Der Diatransport wird jeweils im Dunkelzustand des entsprechenden Projektors vorgenommen.

NUN WIRD'S AUTOMATISCH

Das erste Programm ermöglicht, eine bestimmte Anzahl von Dias immer mit der gleichen Überblend- bzw. Standzeit vorzuführen. Das zweite Programm erlaubt, eine ganze Diaserie mit wechselnden Überblend- und Standzeiten vorzuführen. Die Zeiten werden hierbei in Datazeilen abgelegt und werden mit dem Programm abgespeichert. Die Zeiten hierfür werden in ganzen Zahlen eingegeben, die in etwa Sekunden entsprechen. Durch diese Methode ist es möglich, später weitere Dias ein- oder umzusortieren. Auch kann der Diatitel mit eingegeben werden, der dann jeweils auf dem Monitor sichtbar wird. Ebenso wird auf dem Monitor die Nummer des jeweiligen Dias angezeigt

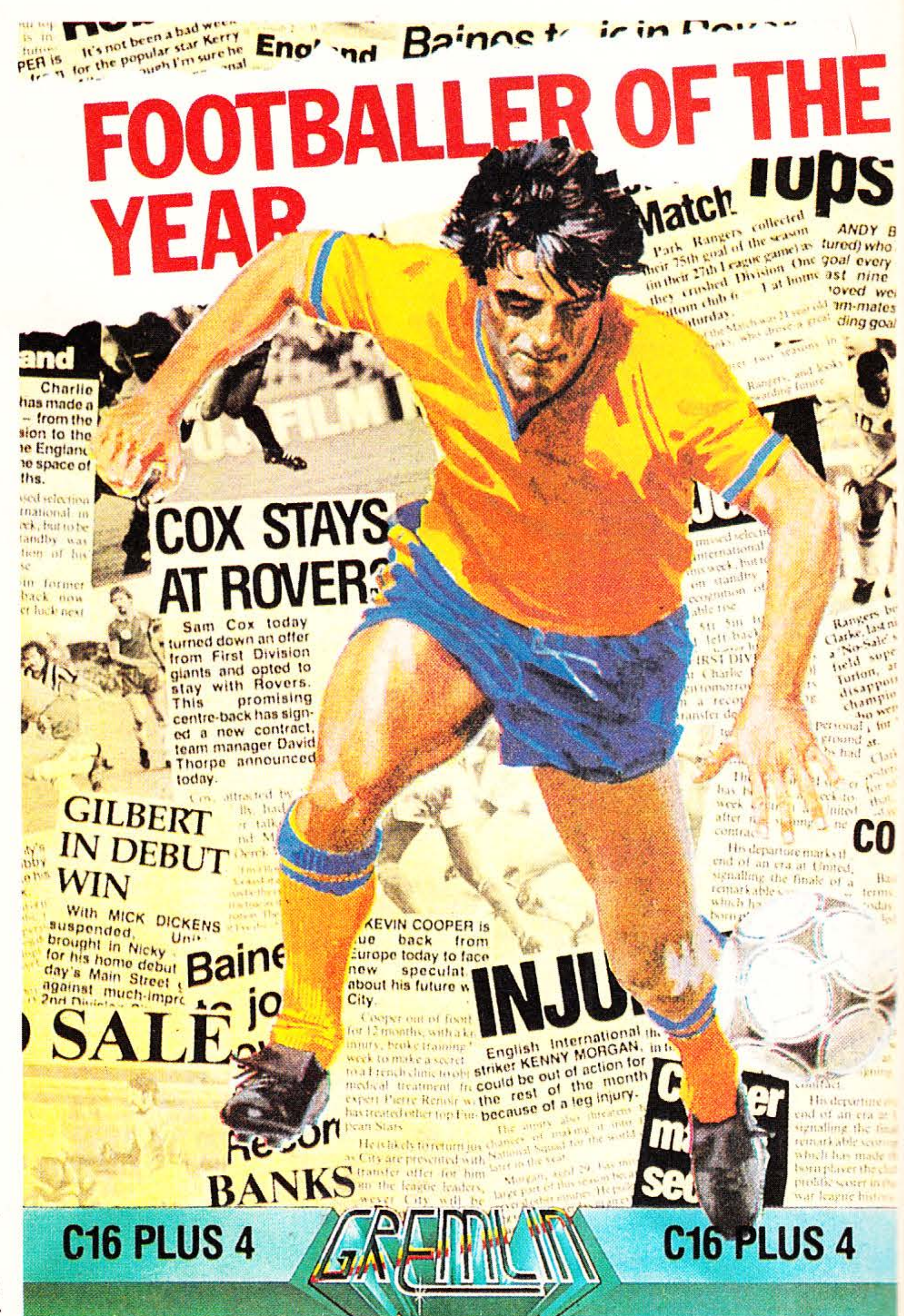
Bitte lesen Sie weiter a. S. 131 

WER IST SCHON BECKENBAUER GEGEN SIE?

Fußballstars, in den Himmel gehoben oder in den Boden gestampft, das ist die Normalität. Und natürlich wünscht sich jeder, ein Beckenbauer oder Netzer zu werden. Jetzt ist es möglich. Mit einem neuen Spiel, über das wir hier berichten.

FOOTBALLER OF THE YEAR

Bei diesem Spiel von Gremlin Graphics haben Sie die Chance, in den obersten Fußballerhimmel zu kommen. Sie können durch den geschickten Einsatz Ihrer Talente einen unaufhaltsamen Aufstieg vom unerfahrenen Anfänger zum ausgebufften Fußballprofi erreichen.
Die Auszeichnung „Fußballer des Jahres" ist die Krönung.
Das Programm gibt es auf Kassette zu 11,90 DM. Auf der Rückseite befindet sich die PLUS4-Version, auf der Vorderseite eine abgemagerte Version für den C16 ohne Erweiterung. Uns interessiert nun erst die PLUS4-Version. Auf die Unterschiede kommen wir später zurück. Wir laden das Spiel mit "LOAD" und sehen, daß sich einiges am Bildschirm tut. Neben "Footy+4" steht nun "Novaload" dort zu lesen und das Flackern des Bildschirmrandes deutet an, daß nun ein Turbolader das Laden des Spieles übernommen hat, übrigens mit Autostart, wie an der Abfrage "DO YOU WISH TO EDIT THE TEAMS (Y/N)?" erkenntlich.
Es lassen sich hiermit für alle Ligen (Divisions) die Namen der Teams ändern. Wer nicht unbedingt seinen Heimatverein in den Tabellen sehen will und sich mit den angebotenen Mannschaften begnügt, kann gleich mit dem Spiel beginnen. Es erscheint nach dieser Entscheidung ein Tornetz mit diversen Informationen über die Macher des Spieles. Außerdem ertönt ein ca. 1 Minute langes Musikstück mit allerhand Variationen, welches für einen PLUS4, dessen unzulängliche Klangmöglichkeiten hinreichend bekannt sind, außergewöhnlich gut gelungen ist. Nach diesem Kunstgenuß wird gefragt, ob man ein bereits begonnenes und abgespeichertes Spiel zum Weitermachen erneut laden will. Dieses ist auch ein Wort, welches sich hören läßt. Am Anfang bleibt uns jedoch vorerst nichts anderes übrig, als wirklich gänzlich am Anfang als völlig unerfahrener Fußballspieler mit 17 Jahren, 5000 Pfund (es ist ein englisches Spiel) und 10 Torkarten zu beginnen. Nach Eingabe des Namens müssen wir uns entscheiden, mit welcher Liga wir beginnen wollen. Die Anleitung empfiehlt, mit der leichtesten zu beginnen, in unserem Falle wäre dies die Nummer 4. Steigen Sie am Anfang zu hoch ein, so werden Sie vielleicht automatisch nach unten wandern. Anschließend können Sie unter 22



Mannschaften dieser Division diejenige auswählen, die Ihnen am sympathischsten ist. Aber keine Angst, bis Sie in die Nationalmannschaft aufgenommen werden, vergeht noch viel Zeit (ich habe es bis heute nicht geschafft)!

ENGLISCHKENNTNISSE SIND GEFRAGT

Wenn man nicht zumindest rudimentäre Kenntnisse in Englisch besitzt, dürfte man mit diesem Programm nicht gerade sehr glücklich werden, zumal eine mickrige deutsche Anleitung in der Größe von 4 Kassettenoberflächen dieses Manko auch nicht auszugleichen vermag. Jedoch mit ein wenig Ahnung dürfte das Spiel nicht schwerfallen.

GEGLÜCKTE KOMBINATION VON AKTION MIT ÜBERLEGUNG UND GLÜCKSSPIEL?

In der PLUS4- und C-16-Version liegt das Spiel als „Strategiespiel" vor. Man hat einfach kurzerhand den Aktionsteil, in welchem man mit Hilfe des Joysticks Elfmeter schießen oder gar gegnerische Spieler ausdribbeln konnte, durch Meldungen, wie "You head the Ball into the net" (Kopfballtor) ersetzt und damit dem Spiel wohl einiges von seinem Reiz genommen. Aber auch als Strategiespiel bleibt es interessant.
Wie gesagt, Sie starten mit 17 Jahren, 5000 Pfund Startkapital und 10 Torkarten Ihre Karriere. Die Torkarten haben eigentlich die größte Bedeutung für den Spieler. Jedesmal, wenn er eine Torkarte „opfert", kann er aktiv bei einem Spiel mitmischen. Eine Torkarte kann zu ein bis drei Versuchen berechtigen, ein Tor zu schießen. Sie müssen bei jedem anstehenden Spiel neu entscheiden, ob Sie eine Torkarte dafür opfern wollen oder nicht.

TOREKARTEN SIND SEHR TEUER – MIT BEDACHT VERWENDEN!

Ich habe nicht herausgekriegt, ob jede Torekarte ihren festen Wert hat, oder ob der Spieler einen Vorteil daraus ziehen kann, wenn er bei Spielen, bei denen ihm nur eine oder zwei Torchancen angeboten werden, dankend die Verwendung der Torekarte ablehnt. Ärgerlich ist es vor allem, wenn man drei Torchancen hatte und alle drei sind danebengegangen.
Dieses Toreschießen ist die einzige Chance des Spielers, Erfahrung und Statusverbesserung zu erzielen. Leider sind die Torekarten ziemlich teuer und müssen mit Bedacht verwendet werden.

DAS KONTROLLMENÜ

Im Kontrollmenü sind verschiedene grafische Symbole dargestellt, die unterschiedliche Informationen liefern oder Aktionen erlauben. Der Spieler wählt den entsprechenden Menüpunkt durch das Ansteuern eines Pfeiles auf das Symbol und anschließendes Drücken der Feuer- oder Spacetaste.
Sie können sich über den Tabellenplatz Ihrer Mannschaft informieren (dies ist sehr wichtig für den Aufstieg in die nächsthöhere Liga am Saisonende). Auch die Mannschaftsmoral wird nicht verschwiegen. Sinnigerweise wird sie immer besser, je höher der Mannschaftstabellenplatz ist. Außerdem werden noch diverse Punktestände aufgelistet und die persönliche Torbilanz des Spielers. Diese ist für den Spielerstatus offenbar ausschlaggebend.
Ein zweiter Menüpunkt gibt über den Zustand des Spielers gezielt Auskunft. Sein Erfahrungsstand, sein Status, die Liga, in der er sich befindet und sein Einkommen (soll ich lästern?) wird teils in Balkendiagrammform dargestellt. Zwei Menüpunkte regeln das Laden und Speichern von Spielzuständen und das Beenden eines Spiels. Sie sehen, das Speichern des Spielzustandes erinnert direkt an Adventures.

DIE AKTIONSPUNKTE

Schließlich gibt es noch drei „Aktionsmenüpunkte".
Einer steht für Spielen, wie das geht, ist oben schon aufgeführt worden. Sie können mitspielen oder nicht, die Mannschaft spielt auf jeden Fall. Nach dem Spiel kommt in Fernschreibermanier das Ergebnis „getickert".
Das zweite Aktionsfeld ermöglicht Ihnen, sogenannte Transferkarten zu kaufen. Sie geben dem Spieler die Chance, den Verein zu wechseln.
So ein Wechsel hat verschiedene Vorteile wie Status-, Liga- oder Einkommensverbesserungen. Nachdem die Transferkarten unsinnig teuer sind und öfters unverhofft als „Ereignis" auftreten und außerdem keine Garantie für einen Transfer bieten, ist der Kauf wohl nur für einen bereits gut verdienenden Profi interessant.
Die „Ereignisse": Während Sie so vor sich hinspielen, verdunkelt sich ab und zu der Bildschirm und ein Ereignis passiert. Diese Ereignisse erstrecken sich von Transferkarten über Torkarten, vom Roulettegewinn bis zu einem kostenlosen Probeschuß auf das Tor, von geklauten Schecks, kaputten Uhren bis zum Rippenbruch. Was genervt hat, war die häufig kommende Meldung: „Der Nationalmannschaftsausschuß sucht noch Spieler . . ." und nach einer kurzen Pause: "Aber Sie wurden nicht gewählt, weil Sie nicht gut genug sind . . .", hahaha!
Wenn Ihnen die von alleine vorkommenden Ereignisse nicht genügen, so können Sie auch beliebig oft Karten vom Stapel zu einem Preis von 200 Pfund kaufen. Dies ist vor allem dann empfehlenswert, wenn der Spieler schon weiter vorgedrungen ist und sein Geldbestand hartnäckig um Null herum bleibt. Die Torkarten, die unbedingt nötig sind zum Vorwärtskommen, steigen nämlich schneller im Preis (pro Liga), als das Einkommen des Spielers. Ohne die Ereigniskarten wäre ich nie zu genug Geld gekommen, um Torekarten kaufen zu können. Außerdem gibt es ab und zu bei den Ereignissen bis zu vier Torkarten geschenkt!

DER SPIELABLAUF

Gespielt wird wie im richtigen Leben jeweils eine Saison lang. Es besteht die Möglichkeit, an verschiedenen Cups teilzunehmen. Die nutzte ich aber mangels Torkarten lieber nicht. Wenn Sie Ihre Mannschaft nicht aktiv unterstützen, wird sie nicht vorwärtskommen. Sie können zwischen den einzelnen Spielen beliebig lang (solange das Geld reicht) Ereigniskarten und Transferkarten kaufen und Ihren und den Status der Mannschaft betrachten. Vorwärts geht es nur, wenn Sie die Spieloption wählen. Dort können Sie zunächst ein bis neun Torekarten kaufen, wenn Sie wollen. Anschließend werden ein oder mehrere anstehende Spiele aufgelistet. Zu jedem Spiel wird ein Torkartenwert angegeben, der sich zwischen ein bis drei Torschüssen bewegt. Sie können nun jeweils entscheiden, ob Sie für dieses Spiel eine Torkarte opfern wollen oder nicht. Haben Sie eine geopfert, so können Sie das Ergebnis positiv beeinflussen (wenn Sie das Tor treffen!). Nach den Torschüssen kommt der Fernschreiber und dann wieder das Kontrollmenü. Dieser Kreislauf setzt sich fort bis zum Saisonende, bei dem über den Aufstieg entschieden wird. Mir ist es zum Beispiel passiert, daß ich gleich



nach dem Aufstieg von einer anderen Mannschaft weggekauft wurde. Ein anderes Mal wurde ich aus einer schlechten Mannschaft der ersten Liga vom Tabellenführer der dritten Liga weggekauft und verlor einiges an Einkommen. Dafür stieg mein Status um fünfzig Prozent! (Und nur der Status scheint zu zählen.) Leider kann ich nicht mit Informationen dienen, wie die Spielentwicklung nach mehreren Wochen Spielzeit vorwärtsschreitet und was für Überraschungen dann noch versteckt sind. Das muß jeder selbst ausprobieren!

TIPS UND TRICKS

Torkarten sind das "A" und "O" des Spiels. Sie können sich nur durch Toreschießen verbessern. Nachdem aber die Karten teuer sind, sollte man sich schon eine gewisse Strategie zurechtlegen. Am Anfang einer Saison müssen Tore geschossen werden, soviel möglich ist. Steht die Mannschaft dann an der Tabellenspitze, kommen die „Jungs" auch etliche Runden ohne Ihre Hilfe aus und Sie können Geld und Torekarten für die nächste Saison aufsparen. Der zweite Trick ist, am Ende der Saison (der Aufstieg droht vielleicht) möglichst viele Torekarten zu kaufen. In jeder nächsthöheren Liga verdoppelt sich der Preis der Torekarten annähernd, nicht aber Ihr Einkommen. Torekarten sind also wie Aktien oder Pfandbriefe. Wenn Sie wenig Geld bar haben, können Sie auch oft Ereigniskarten kaufen, denn weniger als Null Pfund können Sie nicht haben. So kann der Spieler die Vorteile der Ereigniskarten nutzen, ohne die Nachteile in Kauf nehmen zu müssen. Bei den Ereignissen können Sie übrigens oft zwischen drei mehr oder weniger guten oder schlechten Möglichkeiten auswählen. Der Cursor hüpft dabei immer schneller werdend von einer Möglichkeit zur anderen und bleibt stehen, sobald Sie die Feuer- oder Leertaste drücken. Das heißt, je früher Sie drücken, desto eher können Sie das Ergebnis bestimmen.

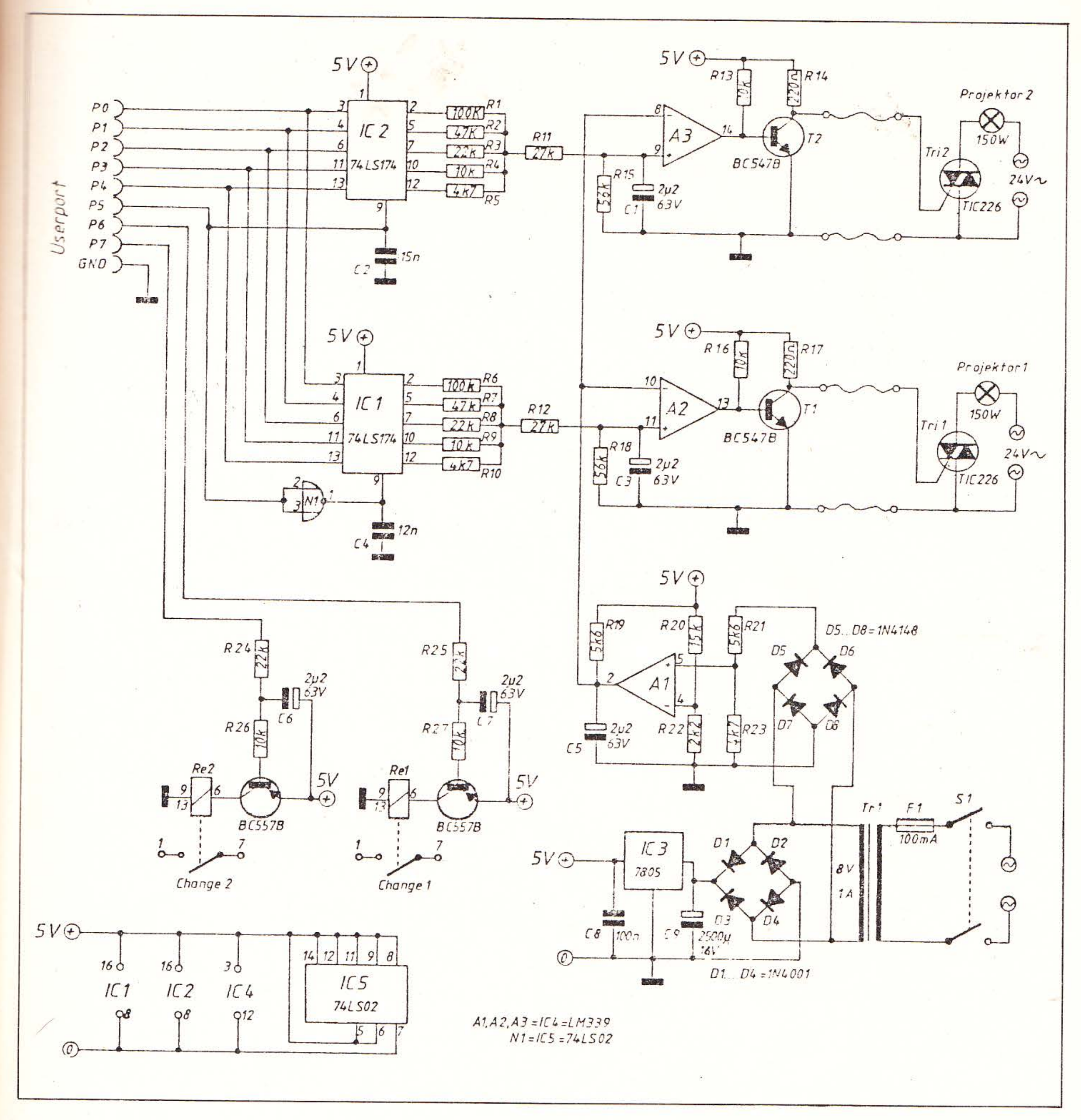
ABGESPECKTE C16-VERSION

In der C16-Version gibt es nicht die Möglichkeit, das Spiel zu speichern. Sie haben so auch nicht wochenlang Zeit, das Ziel „Fußballer des Jahres" zu erreichen. Sie müssen also schon wirklich enorm viel Glück haben, wenn Sie es in so kurzer Zeit schaffen wollen. Bildschirmgrafik oder ansprechende Tabellen wurden durch etwas weniger aufwendigere Textausgabe ersetzt. Anstatt der Balkengrafik, die die Einschätzung als Fußballer beim Plus4 wiedergibt, haben wir nun nur noch den Ausdruck von Zahlenwerten vorliegen, ebenso ist es mit der Anzeige des Tabellenplatzes. Wir haben eine einfache Menüsteuerung, mit der wir durch Drücken der Tasten 1 – 6 die entsprechenden Menüpunkte anwählen. Beim Plus4 dagegen war es eine Anwahl über Bildschirmicons ganz auf die Art und Weise, wie dies in der Workbench eines Amigas geschieht.

FAZIT

Wenn eine Spielidee gut ist, so ist sie es auch noch, wenn man die äußere Gestaltung auf ein Minimum reduziert. Es gibt sehr viele Spiele, die nur durch eine aufwendige Grafik überhaupt erst interessant werden, weil die Spielidee zu einfallslos ist. Es gibt hunderte von Spielen deren Idee es lediglich ist, auftauchenden Hindernissen auszuweichen oder dieselben auch eventuell zu beseitigen (z.B. durch Abschießen), während dabei noch irgendeine Aufgabe zu lösen ist. Oft besteht diese lediglich darin, lange genug durchzuhalten, oder einfach eine bestimmte Strecke zurückzulegen. Daß es hier anders ist, ist erfreulich, daß die Grafik aber ganz zu kurz kommt, ein wenig schmerzlich.
Versöhnlich stimmt uns hingegen der Preis von 11,90 DM, der im Vergleich zur MSX-Version (ca. 30 DM) sehr niedrig ist, und zum Kauf einlädt.

Helmut Maiweg u. A. Mittelmeyer



Fortsetzung von Seite 127

1000 Data X,Y,"Dia Name"

Wobei für X die Überblendzeit und für Y die Standzeit einzusetzen sind.

Sollten noch irgendwelche Fragen auftauchen, wenden Sie sich vertrauensvoll an die Redaktion.

B.W.

Stückliste

R9,R18	= 220 Ohm
R5,R14,R27	= 4,7 kOhm
R4,R8,R13,R17, R20,R22	= 10 kOhm
R3,R12,R19,R21	= 22 kOhm
R2,R11	= 47 kOhm
R1,R10	= 100 kOhm
R6,R15	= 27 kOhm
R7,R16	= 56 kOhm
R23,R25	= 5,6 kOhm
R24	= 15 kOhm
R26	= 2,2 kOhm
C1,C3,C5,C6,C7	= 2,2uF/63V
C9	= 2500uF/ 16V
C4	= 12 nF
C2	= 15 nF
C8	= 100 nF
D1...D4	= 1N4001
D5...D8	= 1N4148
T,T2	= BC547B
T3,T4	= BC557B
Tri1,Tri2	= TIC226
IC1,IC2	= 74LS174
IC3	= 7805
IC4	= LM339
IC5	= 74LS02

Re1,Re2
= Reed-Relais 1301-054-5V (Günther)

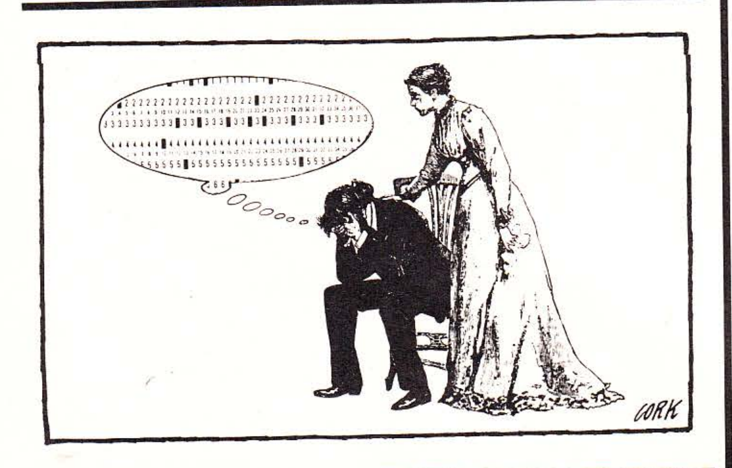
Tr1
= Netztrafo 8V eff/1A sek.

F1
= Feinsicherung 0,1A träge

S1
= Netzschalter 2pol.

DIA-SHOW

Mit diesem Listing liefern wir die Treibersoftware zu unserem Bauvorschlag „Diaprojektor-Steuerung". Nur mit diesem Programm können Sie unseren Umbau auch richtig betreiben.





COMMODOREWELT – DIE UNABHÄNGIGE

```
10 rem diashow listing ==========p4
20 rem (c) by    commodore welt   ==
30 rem ==============================
40 rem (p) by                     ==
50 rem       Armin Kuefner        ==
60 rem                            ==
70 rem version 3.5  40z/ascii     ==
80 rem plus/4 + 1530/41/51        ==
90 rem ==============================
95 gosub 60000
100 cl$=chr$(147):cd$=chr$(17)
110 z=64799
120 pokez,223:pokez,255
130 color0,1:color1,2:color4,1
140 gosub160
150 goto240
160 printcl$cd$"--------------------
--------------------"
170 print" *************************
**************** "
180 print" "rn$"...................
...................."rf$" "
190 print" "rn$"........* computer-
dia-schow *........"rf$" "
200 print" "rn$"...................
...................."rf$" "
210 print" *************************
**************** "
220 print"-------------------------
---------------"
230 printtab(20)"(c)by armin kuefne
r":return
240 printcd$cd$"..wieviele bilder s
ollen vorgefuehrt"
250 printcd$"..werden";
260 inputbx%
270 printcd$"..wieviele sekunden so
ll jedes bild"
280 printcd$"..stehen bleben";
290 inputs%
300 printcd$"..wieviel sekunden sol
len die ueber-"
310 printcd$"..blendungen dauern";
320 inputu%
330 gosub160
340 printcd$cd$cd$tab(5)"(1)..proje
ktoren einstellen"
350 printcd$cd$tab(5)"(2)..direkt a
nfangen"
360 getkeya$:ifa$<>"1"anda$<>"2"the
n360
370 ifa$="1"thengosub650
380 gosub850
390 rem ***************************
*****
400 rem ******** hauptprogramm ****
*****
```

```
410 rem ******************************
*****
420 pokez,x:pokez,y
430 fort=1to(u%*65):next
440 x=x-1:y=y+1
450 ify<>223then420
460 printcl$cd$cd$cd$cd$cd$tab(16)"
bild"b%
470 b%=b%+1
480 gosub610
490 fort=1to(s%*770):next
500 ifb%=bx%+1then1040
510 pokez,x:pokez,y
520 fort=1to120:next
530 x=x+1:y=y-1
540 ifx<>255then510
550 printcl$cd$cd$cd$cd$cd$tab(16)"
bild"b%
560 b%=b%+1
570 gosub630
580 fort=1to(s%*770):next
590 ifb%=bx%+1then1040
600 goto420
610 pokez,159:fort=1to200:next
620 pokez,223:pokez,224:return
630 pokez,127:fort=1to200:next
640 pokez,192:pokez,255:return
650 rem ****************************
**
660 rem ** projektoren einschalten
**
670 rem *** und erste bilder ein- *
**
680 rem ********** legen **********
**
690 rem ****************************
**
700 pokez,223:pokez,255
710 printcl$cd$cd$" projektor 1...b
ild einlegen"
720 pokez,192:pokez,255
730 printcd$" fertig ?..< leertaste
 druecken >"
740 getkeya$:ifa$<>" "then740
750 printcl$cd$cd$" projektor 2...b
ild einlegen"
760 pokez,223:pokez,224
770 printcd$" fertig ?..< leertaste
 druecken >"
780 getkeya$:ifa$<>" "then780
790 printcl$cd$cd$" projektoren aus
richten"
800 pokez,192:pokez,224
810 printcd$" fertig ?..< leertaste
 druecken >"
820 getkeya$:ifa$<>" "then820
830 pokez,223:pokez,255
840 return
850 rem ****************************
***
860 rem ******** titelbild ********
***
870 rem ****************************
***
880 printcl$cd$cd$cd$cd$
890 printtab(16)"computer"cd$cd$
900 print"..d i a s c h o w "
910 fort=1to(s%*770):next
920 x=255:y=223:b%=1
930 pokez,y:pokez,x
940 y=y-1
950 fort=1to(u%*65):next
960 ify<>192then930
970 printcl$cd$cd$cd$cd$cd$tab(16)"
bild"b%
980 b%=b%+1
990 fort=1to(s%*770):next
1000 return
1010 rem **************************
****
1020 rem ********** ausblenden ****
****
1030 rem **************************
****
1040 printcl$cd$cd$cd$cd$cd$
1050 printtab(12)rn$"..."c3$" "c3$c
3$c3$" "c3$".."c3$c3$"..."rf$
1060 printtab(12)rn$" "c3$c3$c3$"..
"c3$c3$" "c3$" "c3$" "c3$" "c3$c3$r
f$
1070 printtab(12)rn$".."c3$c3$" "c3
$" "c3$" "c3$" "c3$" "c3$".."c3$rf$
1080 printtab(12)rn$" "c3$c3$c3$" "
c3$c3$".."c3$" "c3$" "c3$" "c3$c3$r
f$
1090 printtab(12)rn$"..."c3$" "c3$c
3$c3$" "c3$".."c3$c3$"..."rf$
1100 ifx=224then1150
1110 pokez,x:pokez,y
1120 y=y+1:fort=1to(u%*65):next
1130 ify<>223then1110
1140 end
1150 pokez,x:pokez,y
1160 x=x+1:fort=1to(u%*65):next
1170 ifx<>255then1150
1180 end
60000 rem nachspann ==============
60010 rem * farbcodes/steuercodes *
60020 rn$=chr$(018):c3$=chr$(029)
60030 rf$=chr$(146)
60040 return
60050 rem ========================
60060 rem 060671 bytes memory    ==
60070 rem 003660 bytes program   ==
60080 rem ========================
```

TIPS & TRICKS FÜR ALLE 16er!



SOFTWARE-SPEICHER-ERWEITERUNG IM GRAPHIK-MODUS

Hat man die hochauflösende Grafik eingeschaltet, so wird der normale Textbildschirm meist nicht gebraucht. Beschränkt man ihn mit PRINT CHR$(27)+"T" auf eine Zeile, so kann man den Rest des Textspeichers und Farbspeichers (Handbuch S. 228) für Daten nutzen oder durch Heruntersetzen des BASIC-Anfangs den BASIC-Speicher um ca. 980 Byte erhöhen. Dieses kleine Hilfsprogramm (Memoryextension) belegt f1 so, daß nach Tastendruck der Basic-Anfang heruntergesetzt sowie das gewünschte Programm nachgeladen und gestartet wird.

MULTICOLOR IM HIRES-MODUS

Mit 'POKE 65286 OR 64' kann man den Multicolormodus einschalten und mit 'POKE 65286 AND 191' wieder ausschalten. Multicolor bedeutet hier, daß die Zeichen ihren eigenen Farbhintergrund haben, der vom sonstigen Hintergrund verschieden sein kann. Dies funktioniert jedoch nicht für Text in der hochauflösenden Grafik. Aber auch hier gibt es einen Trick. Wird mit 'COLOR O,F, L' die Hintergrundfarbe geändert, während man sich mit GRAPHIC1 im HIRES-Modus befindet, so wird der Hintergrund erst eingefärbt, wenn der Bildschirm mit SCNCLR gelöscht wird. Wird jedoch vorher mit der CHAR-Funktion Text geschrieben, so hat dieser Text einen eigenen Farbhintergrund, und zwar den, der nach dem Bildschirmlöschen entstehen würde. Man kann so also im Prinzip jedem Zeichen einen eigenen Farbhintergrund geben, indem man vor dem CHAR-Befehl mit COLOR O, F, L (F=1 bis 15, L=1 bis 7) den Zeichenhintergrund festlegt. Man hat damit also mehr Möglichkeiten als mit dem normalen Multicolormodus. Zur Demonstration tippen Sie mal das Beispielsprogramm 'MULTICOLOR DEMO' ab und probieren es aus. Sie sehen, daß nicht nur alle Farben (und jede Luminanz) möglich sind, sondern auch Grafikzeichen, was im normalen Textmodus in Multicolor nicht geht. Dort stehen nur 64 Zeichen zur Verfügung.

TIPS UND TRICKS ZU SCRIPT/PLUS: SCRIPT/PLUS ALS DATEIVERWALTUNG

Viele Besitzer dieses Textverarbeitungsprogramms wissen nicht, daß sie sich damit ohne großen Aufwand eine Dateiverwaltung aufbauen können. Mit SCRIPT/PLUS kann man leicht Formbriefe erstellen, und die Adressen und andere Einträge in sogenannten Fill-Files ablegen, um sie beim Ausdrucken in den Formbrief einzufügen. Dies kann man sich zunutze machen, um beliebige Daten abzuspeichern und nach bestimmten Kriterien abzusuchen.

Anlegen des Fill-Files:

Jeder Block muß gleichviele Einträge haben. Der Beginn des Blocks wird mit 'ESC b RETURN' gekennzeichnet (es erscheint ein kleines Rechteck). Jetzt geben Sie die einzelnen Einträge ein. Jeder Datensatz wird mit RETURN abgeschlossen. Er kann ruhig über mehrere Zeilen gehen. Direkt darunter kann man dann den zweiten Block mit 'ESC b RETURN' beginnen usw. Beispiel eines Terminkalenders:

'ESC b RETURN'
28
März
2000
Neue COMMODORE WELT kaufen!!

Dies sind 4 Datensätze im Block. Hat man alle Blocks geschrieben, speichert man das Ganze ab. Es ist empfehlenswert, seine Daten in verschiedene,durch den Namen logisch gekennzeichnete Files aufzuteilen, was die Sucharbeit erleichtert.

Suchen von Daten:

Dazu verwendet man Labels. Diese erzeugt man mit 'ESC SHIFT/b RETURN'. Auf die Frage 'Variable Block Label (a–z)' gibt man b ein und danach wieder RETURN. Dies wiederholt man sooft, wie man Datensätze in einem Block hat (mit c, d, e...).
Bei kurzen Datensätzen ist es sinnvoll, als Format 'rm40:t120:p120' in der ersten Zeile anzugeben. Jetzt kann man diesen Suchfile abspeichern, oder gleich Daten auflisten. Gibt man 'ESC o f v' ein, so werden an die Stelle der Labels die Daten des ersten Blocks eingetragen. Mit zweimal C erscheint der nächste Block. Der Sinn einer Dateiverwaltung ist aber vor allem, sich nur bestimmte Daten auszuwählen. Will man im oben erwähnten Terminkalender alle Termine für den März wissen, so schreibt man hinter c im zweiten Label '=März'. Mit 'ESC o f v' erscheinen jetzt nur Block(s), die als zweites Label 'März' haben. Dabei kann man auch? oder * als Joker verwenden (s. Handbuch). Mit 'ESC SHIFT/v kann man die Einträge auf dem Bildschirm wieder löschen. Auf diese Weise kann man nicht nur Termine, sondern auch seine Schallplatten oder Bücher sowie wichtige Adressen mit dem C16 erfassen und bestimmte Daten leicht wiederfinden.

SCRIPT/PLUS ALS MASKENGENERATOR

Warum sollten Sie sich die Mühe machen, einen Maskengenerator zu programmieren oder abzutippen, wenn das Werkzeug dazu schon hinten im Modulschacht Ihres C16/116 oder Plus4 steckt?
Die Tatsache, daß SCRIPT PLUS mit sequentiellen Files arbeitet, kann man sich zunutze machen, um Bildschirmmasken mit dem Textsystem zu entwerfen, die man dann auf einfache Weise von Band oder Diskette einlesen kann.
Um eine Maske zu entwerfen, gibt man im Eingangsmenü als Textwidth

Multicolor-Demo

```
10 rem ** multicolor demo        **
20 poke65286,91:rem multicolor ein
30 poke65301,0      :rem schwarz
40 poke65302,16*2+1:rem weiss
50 poke65303,16*3+2:rem rot
60 poke65304,16*4+3:rem cyn (lum=4)
70 t$="commodore welt":rv$=chr$(18)
:ro$=chr$(146)
80 color1,2,6:printchr$(147):fors=1
to14
90 char,s,1,mid$(t$,s,1)
100 char,s,2,chr$(asc(mid$(t$,s,1))
+128)
110 char,s,3,rv$+mid$(t$,s,1)+ro$
120 char,s,4,rv$+chr$(asc(mid$(t$,s
,1))+128)+ro$
130 next
140 getkeya$
150 poke65286,27:rem multicolor aus
160 rem *** im hiresmodus ****
170 color0,2,5:graphic1,1:fors=1to1
4
180 forl=1to7:f=l:z=l
190 color0,l,f:char,s,z,mid$(t$,s,1
)
200 nextl:nexts
210 color0,4,0:char,1,l,"SAZXQW"
220 getkeya$:graphic0
```

Boot

```
10 rem * * b o o t * * c w       * *
20 trap30:printchr$(147):directory
30 cr$=chr$(29):cd$=chr$(17):cu$=ch
r$(145):cl$=chr$(157)
40 fori=1to18:r$=r$+cr$:next
50 l$=cl$+cu$+"  "+cl$+cd$+cd$+" "
60 d$="dload"+r$+"      ":a=3077:z=0
70 z$=left$(r$+r$,23)+chr$(130)+"<"
+chr$(132)
80 ifpeek(a)<>34thena=a+40:z=z+1:go
to80
90 char,0,z,z$+l$
100 getkeya$:z=z-(a$=cd$)+(a$=cu$):
ifz<0thenz=0:elseifz>24thenz=24
110 if a$="a" then d$="load
"+r$+",8,1"
120 ifa$<>chr$(13)goto90
130 char,0,z,d$:poke239,6:t=1319
140 poket,145:poket+1,145:poket+2,1
3:poket+3,82:poket+4,213:poket+5,13
```

Memoryextension

```
0 rem ***** memoryextension ******
10 cr$=chr$(13):hk$=chr$(34):cl$=ch
r$(147)
20 input"filename";p$
30 key 1,cl$+"p043,41:p044,12:p0311
2,0:new"+cr$+"dL"+hk$+p$+hk$+cr$+"r
U"+cr$
40 scnclr:print"bitte f1 druecken..
.........."chr$(27)"b";:getkey a$
```

Deutscher Zeichensatz für Script/Plus

```
90 rem * deutscher zeichensatz **
100 rem*fuer script-plus ********
110 a=peek(56)-3:rem basic-ende her
absetzen
120 if a/4 <> int(a/4) then a=a-1:g
oto 120
130 poke 56,a:poke 54,a:poke 55,0:p
oke 53,0:clr
140 a=256*peek(56):print "neue matr
ix ab adresse";a
150 poke 1177,62:rem fuer peek auf
rom schalten
160 for i=0 to 1023:poke a+i,peek(5
4272+i):next
170 rem zeichengenerator ins ram
180 poke 65298,peek(65298)and251:re
m matrix im ram
190 poke 65299,(peek(65299)and3)or
a/256:rem matrixadresse
200 poke 740,a/256:rem matrixadress
e fuer char-befehl
210 poke 2042,0
220 n=7:rem anzahl der zeichen
230 for i=1 to n:read c$:printchr$(
147);c$:c=peek(3072)
240 for x=0 to n:read b:poke a+8*c+
x,b:next:next
250 data "^",0,56,108,120,108,108,1
20,96:rem "ss
260 data "[",0,102,60,6,62,102,62,0
      :rem "ae
270 data "\",0,102,60,102,102,102,6
0,0  :rem "oe
280 data "]",0,102,0,102,102,102,62
,0   :rem "ue
290 data "<",102,24,60,102,126,102,
102,0:rem "Ae
300 data "@",102,60,102,102,102,102
,60,0:rem "Oe
310 data ">",102,0,102,102,102,102,
60,0 :rem "Ue
```

Achtung! Das Zeichen in Zeile 250 (^) ist der Pfeil nach oben, in Zeile 270 (\) das Pfund-Zeichen!!! »

40 an. Im EDIT-Modus kann man nun die Maske genauso entwerfen, wie man sie haben will. Ein großer Vorteil von SCRIPT/PLUS ist dabei die Tatsache, daß man auch den größten Teil der Grafikzeichen, die mit der COMMODORE-Taste geschrieben werden, verwenden kann (ein Teil wird als Steuerzeichen für Fill-Files ... gebraucht). Sehr hilfreich ist auch der Block-Range-Befehl 'ESC SHIFT/r'. Will man nämlich einen ganzen Textblock verschieben, weil einem die Optik nicht gefällt, so fährt man mit dem Cursor in die linke obere Ecke des zu verschiebenden Rechtecks, gibt 'ESC SHIFT/r' ein und fährt den Cursor an die linke untere Ecke des Rechtecks. Nach RETURN kann man diesen Block mit den Cursortasten beliebig verschieben, um ihn dann mit RETURN zu fixieren.
Manchmal ist es notwendig, mit 'ESC w' die Wordwarp-Funktion' auszuschalten. Will man den von der Maske benötigten Speicherplatz auf Diskette klein halten, so gibt man in jeder Zeile an der Stelle, wo nur noch Leerzeichen kommen, ein RETURN ein. Dann werden die rechten Leerzeichen nicht mit abgespeichert.

WIE KOMMT DIE MASKE AUF DEN BILDSCHIRM?

Dazu muß im Programm nur eine Zeile eingefügt werden:

```
1000 open2,8,2,p$+",s,r"
   :printchr$(147):fori=
   0to1:get#2,a$:printa$;
   :i=st:next:close2:return
```

Diese Zeile wird dann im Bedarfsfalle mit

```
100 p$="maske":gosub
1000
```

angesprungen. Zeile 1000 sollte noch kurz erklärt werden. Solange st=0 ist, erreicht der Wert für i niemals 1, und die FOR-NEXT-Schleife wird daher solange durchlaufen, bis das File auf der Diskette zuende ist.
Bei Kassettenbetrieb geht man ähnlich vor, nur ersetzt man OPEN 2,8,2, P$+",S,R" durch OPEN 1,1,0,P$ und statt GET#2 setzt man GET 1 ein.

PROGRAMME VON DISKETTE LADEN MIT BOOT

Hier ist ein kleines Programm für Schreibfaule. Es muß als erstes auf der Diskette stehen. Mit COMMODORE-RUN/STOP wird immer das erste Programm geladen und gestartet. Nun erscheint das Directory, welches man wie üblich mit CTRL/S anhalten kann. Ist das Directory beendet oder mit STOP abgebrochen, so erscheint hinter dem ersten Filenamen ein blinkendes Kleiner-Zeichen, welches man mit den Cursortasten rauf und runterbewegen kann. Steht es hinter dem Programm, welches geladen werden soll, braucht man nur noch RETURN drücken. Den Rest macht das Programm. Soll das Programm absolut geladen werden (Maschinenprogramme), so drückt man vor RETURN ein A. Dieses Programm sollte auf jeder Diskette am Anfang stehen und spart wirklich lästige Tipparbeit. Da das Programm so kurz ist, ist es sehr schnell geladen.

WIE LEST MAN AB $8000 AUS DEM RAM?

Dies ist nur in Maschinensprache möglich. Hier ist es sogar relativ einfach. Zuerst muß der Interrupt gesperrt werden (er findet ja im RAM seine Interrupt-Routine nicht) und dann kann man mit einem Schreib-Zugriff (STA) auf $FF3F auf das RAM umschalten. Danach kann man aus dem RAM lesen und schaltet mit einem Schreib-Zugriff diesmal auf $FF3E zurück zum ROM, wobei

```
0 rem = daumenkino korrekturen ====
140 max=int((peek(56)-32)/4)
310 if an<=max then 240:else 150
380 b$=str$(28+hn*4)
450 poke217,24+an*4:sys1662:return
480 poke217,28:fori=1tomax:sys1662:
fort=0tov:next:next:goto480
510 fori=1630to1700:reada:pokei,a:s
=s+a:next:ifs=10813thenprint"ok":el
seprint"fehler in datas":end
520 return
530 data169,0,133,218,169,12,133,21
9,162,4,160,0,177,218,145,216
540 data200,192,0,208,247,230,217,2
30,219,202,224,0,208,236,96,0
550 data169,0,133,218,169,12,133,21
9,120,141,63,255,162,4,160,0,177,21
6,145,218
560 data200,192,0,208,247,230,217,2
30,219,202,224,0,208,236,141,62,255
,88,96
```

```
0 rem *** zeichensatz fuer 12 K **
100 poke55,255:poke56,59:clr:rem sp
eicher begrenzen
110 key1,"p065298,196:p065299,208"+
chr$(13)
120 data162,0,189,0,208,157,0,60,18
9,0
130 data209,157,0,61,189,0,210,157,
0,62
140 data189,0,211,157,0,63,202,208,
229,96
150 fora=832to861:readb:pokea,b:nex
t
160 sys832:poke65298,192:poke65299,
60
170 zs=15360:rem zeichensatz-beginn
```

```
0 rem *** zeichensatz fuer 64 K **
100 poke55,255:poke56,248:clr:rem s
peicher begrenzen
110 key1,"p065298,196:p065299,208"+
chr$(13)
120 data162,0,189,0,208,157,0,248,1
89,0
130 data209,157,0,249,189,0,210,157
,0,250
140 data189,0,211,157,0,251,202,208
,229,96
150 fora=832to861:readb:pokea,b:nex
t
160 sys832:poke65298,192:poke65299,
248
170 zs=63488:rem zeichensatz-beginn
```

der Wert, der in die beiden Speicherstellen geschrieben wird, keine Rolle spielt.
Genau dies macht nun die neue Maschinenroutine. Zudem wurden noch ein paar kleinere Unsauberkeiten behoben. Wenn Sie also Daumenkino schon eingetippt haben, so ändern Sie bitte die angegebenen Zeilen ab.

TIPS AN UNSERE SPIELE-AUTOREN

Immer wieder flattern uns sehr gut programmierte Joystickspiele auf den Tisch, die allesamt den gleichen Fehler haben: Sie wurden von Besitzern einer 64 K-Erweiterung geschrieben und so ausgelegt, daß sie auch auf 12 K laufen sollten. Da das Spiel meist nicht länger als 8 K-Byte ist, wurde der neue Zeichensatz knapp dahinter gesetzt und damit sollte es dann klappen. Nun haben wir damit meist 2 Probleme:
1. Oft bleibt kein Platz zwischen Programmende und Zeichensatzanfang, um die bei uns übliche Umwandlung der unleserlichen Steuercodes in Strings vorzunehmen und den Copyright-Kopf anzubringen.
2. Dieser Zeichensatz wurde nicht geschützt. Dies geschieht normalerweise durch Herabsetzen des Basic-Speicher-Endes in 55/56.
Zusätzlich muß aber auch noch ein CLR-Kommando eingefügt werden, um die anderen Zeiger, wie zum Beispiel den Beginn der Strings, anzupassen. Sonst geschieht nämlich folgendes:
Bei 64 K passiert nichts, da die Strings ja dort beginnen, wo der Zeiger in 51/52 hinzeigt, nämlich nach wie vor am Ende der 64 K. Bei 12 K aber wird nach kurzer Zeit der wunderschöne neue Zeichensatz überschrieben. Daher haben wir allen, die den Wunsch haben, einmal ein Spiel mit verändertem Zeichensatz zu schreiben, eine dazu nötige Routine abgedruckt. Hier liegt der neue Zeichensatz jeweils am Ende der 12 bzw. 64 K. Die Startadresse ist angegeben. Den Beginn eines bestimmten Zeichens findet man, indem man zur Startadresse noch 8*Bildschirmcode (des Zeichens) dazuzählt. In Zeile 100 wird zusätzlich die f1-Taste so belegt, daß beim Drücken der normale Zeichensatz erscheint. Dies ist sehr sinnvoll, da es die wenigsten schaffen, ein Programm völlig fehlerfrei abzutippen und ein Programmabsturz bei geändertem Zeichensatz zu wirren Zeichen auf dem Bildschirm führt. Man kann dies aber auch verhindern, indem man die READY-Routine ändert, wie dies von Ulrich Schäfer im C16-Sonderheft 3/86, S. 26, vorgeschlagen wurde.

WIE PROGRAMMIERT MAN EIN JOYSTICKSPIEL?

Der C16 ist durch sein starkes Basic sehr gut eingerichtet, um Joystickspiele ohne großen Aufwand zu erstellen. In den meisten Fällen wird für Spiele nicht die hochauflösende Grafik verwendet, da die SHAPES zu langsam sind. Um trotzdem eine ansprechende Grafik zu erhalten, wird der Zeichensatz modifiziert und die Männchen, Autos, Hindernisse . . . aus mehreren Zeichen zusammengesetzt. Die Routine zur Zeichensatzverlegung ist ja schon da. Nun wollen wir uns mal überlegen, was noch gebraucht wird, um ein unterhaltsames Spiel zu erstellen.

STUFE 1: DIE IDEE

Zuerst muß die Spielidee vorhanden sein.
Ein UFO-Abschieß-Ballerspiel lockt keinen Hund mehr vom Ofen vor. Das müßte schon in 10 Ebenen mit hervorragender Grafik und schneller Maschinensprache sein, um noch Begeisterung hervorzurufen. Auch Männchen, die in einem Labyrinth Schätze einsammeln, sind schon zu Dutzenden bei uns erschienen. Hier ist also Originalität gefragt. Obwohl nicht besonders einfallsreich (dient ja auch mehr zu Lernzwecken), soll nun die Idee unseres Demospiels erklärt werden:
Das Problem des Handelsreisenden: Wer sich mit Mathematik befaßt, der kennt vielleicht dieses Problem. Ein Handelsreisender soll verschiedene Städte auf dem kürzesten Weg bereisen. Unser Handelsreisender hat nun die Aufgabe, 10 Damen zu besuchen, um (nicht was Sie denken) ihnen Kosmetika zu verkaufen. Dafür stehen ihm 1500,– DM Spesen zur Verfügung. Jeder Kilometer und jede Stunde, die er zur Erledigung seiner Aufgabe braucht, verringert seinen Verdienst. Damit das Ganze nicht zu einfach ist, stehen noch Hindernisse im Weg.

STUFE 2: DIE GRAFIK

Eine hübsche Bildschirmgestaltung ist schon das halbe Spiel. Auch die beste Idee wirkt langweilig, wenn man sich dabei auf den eingebauten Zeichensatz beschränkt. In diesem Beispiel jedoch wurde nur der gefüllte Kreis für den Handelsreisenden, das Herz für die Damen und das #-Zeichen für die Hindernisse verwendet. Meist reicht ein Zeichen nicht aus, um die gewünschte Figur darzustellen. Dann muß man mehrere Zeichen zu einem String zusammenfassen.

STUFE 3: DER BILDSCHIRMAUFBAU

Damit nicht jedes Spiel wie das vorhergehende verläuft, sollte bei der Verteilung der Hindernisse die RND-Funktion eingesetzt werden (S. Zeilen 130–160). Die Zeichen kann man nun auf 3 Arten auf den Schirm bringen. Mit der PRINT-Funktion (hier ist man festgelegt), durch POKEn in den Bildschirmspeicher (dies ist sehr rechenintensiv und daher langsam) oder wie hier mit der CHAR-Funktion. Ein großes Problem stellt dabei jedoch der Rand des Bildes dar. Schreibt man ein Zeichen in die 39. Spalte (man rechnet von 0–39), so wird eine Folgezeile eingefügt, was den Bildschirmaufbau zerstört. Hier kann man natürlich vor jeder CHAR-Anweisung den Wert für XN und YN überprüfen. Es gibt aber noch eine schlauere Lösung, indem man den TRAP-Befehl sinnvoll einsetzt. Der TRAP-Vektor wurde hier auf die Zeile 280 gerichtet, wo die Werte für XN und YN überprüft und korrigiert werden. Dies wird nun nur dann durchgeführt, wenn in der CHAR-Funktion ein unerlaubter Wert steht. Diese Prozedur erspart eine Menge Rechenzeit und das Programm ist schneller.

STUFE 4: DIE BEWEGUNG DER FIGUR

Durch die JOY-Funktion kann die Stellung des Joysticks leicht abgefragt werden. Wie aber kann man dies auf die Figur übertragen? In 200–210 werden auf sinnvolle Weise die logischen Operatoren eingesetzt, um aus der Joystickstellung ein DX und DY zu errechnen. Diese Werte geben zusammen mit der alten X/Y-Position die neue Stellung. Damit wird dann in Zeile 220 geprüft, ob die entsprechende Position im Bildschirmspeicher ein Leerzeichen enthält. Mit dieser Information kann dann weiter verzweigt werden. Entweder man trifft auf das Hindernis – oder die Figur kann bewegt werden. 

Auch hier wird bei Überschreiten des Randes die Korrekturroutine ab 280 angesprungen.

STUFE 5: DER SOUND

Dies ist ja beim C 16 einfach zu programmieren. Wenigstens sollte bei Erreichen eines Zielpunktes ein Ton ertönen, was eine gewisse Erfolgskontrolle darstellt. Auch eine kleine Eingangsmelodie ist sehr effektvoll. Dabei sollte aber immer darauf geachtet werden, daß diese durch Tastendruck unterbrochen werden kann . . .

STUFE 6: DAS SPIELERGEBNIS

Während des Spieles sollte immer eine Anzeige des momentanen Standes erfolgen, damit man sieht, wie weit man sich dem gesteckten Ziel schon genähert hat. Am Ende des Spiels erfolgt die Ausgabe eines Ergebnisses mit einer Angabe des bisher besten Wertes, um eine Wiederholung und Steigerung anzuregen.

STUFE 7: BESCHLEUNIGUNG DES PROGRAMMS

Ein solches Spiel sollte möglichst schnell sein. Daher ist es wichtig, einige Regeln zu beachten:

a) Oft verwendete Routinen sollten ganz vorne oder direkt hinter der Zeile, von der sie mit GOSUB angesprungen werden stehen, selten verwendete gehören nach hinten ins Programm. Bei GOTO und GOSUB wird zuerst geprüft, ob die Sprungadresse größer ist, ansonsten wird von vorne nach dieser Zeile gesucht.

b) Oft verwendete Variablen müssen als erstes definiert werden, auch wenn sie noch nicht gebraucht werden.

c) Mehrere IF-Abfragen in der Reihenfolge ihrer Häufigkeit anordnen, das heißt, zuerst die, welche am öftesten erfüllt ist, so daß die anderen gar nicht erst untersucht werden müssen.

d) Wird eine Zahl (z.B. 3072 für den Beginn des Bildschirmspeichers) öfters verwendet, so lohnt es sich, diese als Variable zu definieren. Schließlich steht 3072 als 4 Byte im Speicher, die einzeln vom Interpreter gelesen, interpretiert und dann am Ende zu einer Zahl zusammengesetzt werden müssen. Es lohnt sich, diese Regeln zu befolgen und das Programm auf Geschwindigkeit zu optimieren.

STUFE 8: EXTRAS

Bei Verwendung der TRAP-Funktion sollte auf andere Weise (als die STOP-Taste) der Ausstieg aus dem Programm möglich sein. Beispiel: Zeile 180 mit der Feuertaste, Zeile 250 mit ESC.

LISTSCHUTZ – WIE?

Jeder, der sein Programm schützen will, sollte wissen, daß es den perfekten Listschutz nicht gibt. Jeder Schutz kann geknackt werden. Es ist nur möglich, es dem Knacker so schwer wie möglich zu machen.
Zunächst muß man sich entscheiden, was man schützen will. Will man das Programm überhaupt vor fremdem Zugriff schützen und es nur selbst verwenden können, so gibt es einen sehr einfachen Trick. Vor dem Abspeichern setzt man den BASIC-Anfang (steht in den Speicherstellen 44/45) herunter, z.B. mit POKE43,200:POKE 44,15. Dann speichert man normal ab. Hierbei werden die Speicherstellen, um die man den BASIC-Anfang heruntergesetzt hat, mit abgespeichert. Wird das Programm normal geladen, so wird es immer an den BASIC-Anfang bei 4096 geladen und beginnt mit völlig unsinnigen Zeichen, die vorher unten am Bildschirm standen. Will man das Programm verwenden, so muß man es mit LOAD"name", 8,1 bzw. ,1,1 laden. Jetzt steht alles wieder am richtigen Platz und kann normal gestartet werden.
Oft will man nur die DATA-Zeilen im Programm vor fremden Blik-

LISTSCHUTZ LEICHTGEMACHT

```
10 rem Handelsreisender ==========16
20 rem (c) 03/87 commodore welt   ==
30 rem ===============================
100 b=3072:v=40:n=32:g=35:h=83:colo
r0,?,4:color4,2,3:vol7
110 e$=chr$(27):trap280:printchr$(2
7)"m"
120 z$="Q":1$=" ":scnclr
130 fori=1to35:xn=int(rnd(1)*39):yn
=int(rnd(1)*24):fork=1to7
140 xn=xn+sgn(rnd(1)-.5):yn=yn+sgn(
rnd(1)-.5)
150
char,xn+1,0,"":char,xn,yn,"#":ne
xt:next
160 fori=1to11:char,rnd(1)*39,rnd(1
)*23+1,chr$(28)+chr$(130)+"S":next:
printchr$(144)chr$(142)
170 x=20:y=10:char,x,y,z$:ifjoy(1)=
0then170:elseti$="000000"
180 j=joy(1):ifj=0then180:elseifj=1
28then380
190 z=z+1:char,0,0,"steps:"+str$(z)
200 dx=-(j>1andj<5)+(j>5)
210 dy=(j<3andj>0)+(j=8)-(j>3andj<7
)
220 xn=x+dx:yn=y+dy:ch=peek(b+v*yn+
xn):ifch<>nthen340
230 ifxn>38thenxn=0
240 char,x,y,1$:x=xn:y=yn:char,x,y,
z$
250 geta$:ifa$=e$thenprintchr$(27)"
1":stop
260 goto180
270 rem ** trap rand **********
280 ifxn<0thenxn=37
290 ifxn>38thenxn=0
300 ifyn<0thenyn=23
310 ifyn>24thenyn=0
320 resumenext
330 rem ** hinderniss *********
340 ifch=gthen180
350 ifch=hthensound1,300,10:sc=sc+1
:char,20,0,"score:"+str$(sc):ifsc=1
0then380
360 goto230
370 rem ** schluss ************
380 printchr$(27)"l"chr$(147)chr$(1
```

en schützen (z.B. in einem Adventure). Auch hier gibt es einen einfachen Trick, bei dem der RENUMBER-Befehl sehr hilfreich ist. Dieser hat nämlich einen kleinen Fehler eingebaut. Er läßt es zu, daß nach dem Neunumerieren Zeilennummern entstehen, die größer als 64000 sind. Diese können weder gelöscht noch editiert oder mit GOTO angesprungen werden. Hat man das Programm ganz fertiggestellt und die DATA-Zeilen befinden sich am Programmende, so wird ein RENUMBER durchgeführt, bei dem die letzte Zeile vor den DATAs die Nummer 63999 erhält.

Hier ein Beispiel:

```
100 fori=1to10:readx
    :printx:next
200 rem ab hier
    geheim!!####
300 data 1,1,2,3,1
400 data 3,3,2,1,2
```

Mit RENUMBER 63999,1, 200 haben alle Zeilen mit Nummern größer 200 jetzt Nummern größer 64000 und können nicht mehr gelöscht oder verändert werden.

Aber noch hat die Sache einen entscheidenden Haken. Bei LIST werden sie trotzdem normal gelistet. Man muß also das Listen vorher abbrechen. Beachten Sie einmal die vier # am Ende von Zeile 200 (nach dem RENUMBER Zeile 63999). Jetzt geben Sie einmal im Direktmodus folgende Zeile ein:

```
fOi=4100to9*i:ifpE(i)
  =35andpE(i+1)=35
  tHpOi+1,27:pOi+2,84:
  pOi+3,27:pOi+4,66
  :elsenext
```

Die vier # werden dadurch in ESC/t und ESC/b umgewandelt. Nach

DAS LISTING BLEIBT IM FENSTER HÄNGEN

einem REM-Befehl werden alle CHR$-Codes kleiner 32 ausgeführt und ESC hat den Code 27. Dies bewirkt, daß bei Erreichen dieser Stelle ein Fenster von der Größe eines Zeichens an der Cursorposition erzeugt wird. Damit ist der Rest nicht mehr erkennbar, obwohl er in diesem Fenster gelistet wird. Ein LIST 64000 erzeugt einen SYNTAX-ERROR. Ein erneutes RENUMBER würde unsere Bemühungen aber zunichte machen. Aber auch dies kann man verhindern. Der RENUMBER-Befehl leistet uns auch hier Schützenhilfe. Wird bei Ausführung des RENUMBER-Befehls eine nichtexistierende Sprungadresse gefunden, so wird diese zu 65536. Bei einem zweiten RENUMBER erzeugt diese Sprungadresse größer 63999 natürlich wieder einen SYNTAX-ERROR. Wir brauchen also in obiges Beispiel vor dem RENUMBER-Befehl nur noch die Zeile 210 einfügen:

210 goto 220

Da 220 nicht existiert, ist damit ein zweites RENUMBER blockiert. Diesen Fenstertrick kann man natürlich beliebig oft im Programm anwenden.

Damit nun aber genug für heute. In der nächsten Folge geht es weiter mit Listschutz Teil 2. Sie können inzwischen das Gelernte ja schon mal anwenden. Aber bitte nicht bei den Listings, die Sie uns einsenden.

```
4)
390 print"Sie haben"int(ti/6)/10" S
tunden gebraucht und"
400 print"sind"z" Kilometer gefahre
n."
410 print: print"Von Ihren 1500,-DM
 Spesen sind daher"
420 re=int(1500-ti/500-z*3)
430 print"nur noch"re",-DM uebrig."
440 ifre>hs then hs=re
450 print:print"Bisher bestes Ergeb
nis:"hs
460 poke239,0:input"noch einmal";q$
:ifq$="j"thensc=0:goto100
```

Beispiel-Programm

WAS IST BESSER, C16 ODER PLUS4?

Bis vor einigen Monaten war dies eine Frage des Geldes. Ein C16 war mit Datasette für 149,– DM zu haben, ein PLUS4 kostete allein schon über 300,– DM. Den C16 konnte man für 99,– DM auf 64KByte aufrüsten und bekam dabei noch SCRIPT/PLUS (ein sehr gutes Textverarbeitungsprogramm) gratis dazu. Jetzt gibt es preisgünstigere Angebote.

PLUS4 inklusive Floppy 1551 für sage und schreibe 399,– DM. Anlaß genug für uns, den C16 mit 64 K einmal mit dem PLUS4 zu vergleichen. Reden wir zunächst von dem, was dem PLUS4 den Namen gab, von der eingebauten Software. Als der PLUS4 entwickelt wurde, war diese Software sicherlich bemerkenswert. Inzwischen sind das Niveau und die Ansprüche gewaltig gestiegen. Die Textverarbeitung wird z.B. von SCRIPT/PLUS um Längen geschlagen, so daß sich viele PLUS4-Besitzer bereits entschlossen haben, das entsprechende ROM gegen das von SCRIPT/PLUS zu ersetzen (siehe CW 2/87). (Vergleicht nur einmal die zu verarbeitende Textmenge.) Beim Textprogramm vom PLUS4 kann man 99 Zeilen zu 77 Spalten bearbeiten. Beim SCRIPT/PLUS sind es zwei voneinander unabhängige Texte mit je 20 000 Zeichen, wobei man sich die Spalten/Zeile einstellen kann.

Von den anderen Unterschieden gar nicht zu reden. Dies wurde schon in vielen Heften ausführlich getan. Ähnlich verhält es sich mit der Tabellenkalkulation und der Datenverwaltung, verglichen mit CALC/PLUS. Aber dies ist ja kein Hinderungsgrund gegen den PLUS4, denn den kann man auch mit der erwähnten Software verbessern. Ansonsten sind von der Hardwareseite beide Rechner fast gleich. Auf beiden laufen die gleichen Programme. Aber es gibt noch einen entscheidenden Vorteil des PLUS4 und das ist der Userport. Der wurde beim C16 abgespeckt und muß durch eine Hardwareerweiterung „nachgerüstet" werden, wenn man ihn braucht. Aber Vorsicht, es gibt zwei verschiedene PLUS4, einen mit und einen ohne Userport. Das 399,– DM-Angebot gilt für einen PLUS4 mit Userport. Das Gehäuse ist niedriger als beim C16 und die Funktionstasten liegen oben, so daß der PLUS4 auch schmaler ist. Die Tastatur ist etwas empfindlicher als beim C16. Man überschribt leicht einmal eine Zeile, wenn die Finger vom vielen Tippen etwas müde werden. Es ist daher oft sinnvoll, mit POKE1344,64 die Wiederholfunktion der Tasten auszuschalten. Die Floppy 1551, welche mit dem PLUS4 mitgeliefert wird, verdient auch noch ein paar lobende Worte. Sie wird mit einem Parallelkabel am Expansionsport betrieben (der zum

Glück durchgeschleift ist) und ist ca. 3mal so schnell wie die Floppy 1541 und wird auch nicht so schnell heiß. Ferner erscheint die ganze Mechanik stabiler und die Floppy initialisiert sich selbständig, wenn eine Diskette eingeschoben wird. Dadurch kann es nicht zu unangenehmen Überraschungen kommen, wenn man zwei Disketten mit der gleichen ID verwendet. Die 1541 denkt nämlich in solchen Fällen, es ist die gleiche Diskette und überschreibt unter Umständen Teile der Diskette. Leider kann man die 1551 nicht am C64 betreiben, die C64-Besitzer müssen für die 1541 immer noch ca. 400,– DM hinblättern. Aber vielleicht kommt einer unserer Leser mal auf den Trick, wie man den C16/Plus4 als RAM-Floppy und als Pufferspeicher für die 1551 oder den Drucker benutzen kann. Dann könnten auch C64-Besitzer von diesem Angebot profitieren. Alles in allem kann man dieses Komplettangebot empfehlen.

PLUS4 TEXTFILES AUF SCRIPT/PLUS

Wenn Sie schon länger einen PLUS4 besitzen, auch schon einiges an Text damit geschrieben haben und sich nun entschlossen haben, SCRIPT/PLUS zu verwenden, so sind Ihre alten Texte nicht verloren. SCRIPT/PLUS arbeitet mit dem wesentlich vielseitigeren ASCII-Code, im Gegensatz zum PLUS4, der den Bildschirmcode verwendet. Das kleine Listing 'text plus4 auf script/plus' wandelt den Code um. Sie müssen nach dem Starten nur den Namen des PLUS4-Files auf Diskette angeben und den neuen File mit SCRIPT/PLUS etwas nachbearbeiten.

Will man ein professionelles Programm erstellen, so ist es manchmal sehr nützlich und eindrucksvoll, wenn bestimmte Eingaben, wie zum Beispiel Druckerparameter, nicht einzeln mit einer Inputanweisung abgefragt werden, sondern in einer Eingabemaske alle Werte schon vorgegeben sind und der Benutzer nur noch die ändern muß, die verändert werden müssen. Dabei sollte er das Eingabefenster nicht verlassen können. Mit dem Beispielprogramm 'inputroutine mit fenster' ist dies alles verwirklicht. Zunächst werden die Daten aus den DATA-Zeilen ausgegeben und die Fenster definiert. Mit den Cursortasten kann man sich innerhalb der Eingabefenster bewegen und von Fenster zu Fenster springen. In Zeile 1070 wird durch ESC/m das Scrollen verhindert, so daß die Eingabe nicht verschwindet. Das Listing erklärt sich weitgehend von selbst. Sie müssen nur darauf achten, daß in den DATA-Zeilen die Texte inklusive Wertvorgabe richtig dimensioniert werden. Zum Beispiel muß in diesem Beispiel hinter MAI genug Platz gelassen werden, damit auch DEZEMBER geschrieben werden kann. Versuchen Sie doch einmal das Eingabefeld um eine Größe zu erweitern. Zum Beispiel können Sie noch die Stunden einfügen. Dazu in Zeile 1010 n auf 4 erhöhen und eine vierte Data-Zeile einfügen. Besonders wichtig sind natürlich die Zeilen 1160–1170. Hier wird der Bildschirm als Eingabekanal geöffnet und die Zeichen im ASCII-Format mit GET# vom Bildschirm gelesen. Dies ist natürlich wesentlich einfacher als die Umrechnung der Bildschirmcodes aus dem Bildschirmspeicher.

```
0 rem ******************************
*
1 rem ** inputroutine mit fenster *
*
2 rem ******************************
*


10 fori=1to40:cr$=cr$+chr$(29):next
:trap50
20 gosub1010:rem aufruf der routine
30 rem ausgabe der neuen werte, hie
r wird das programm fortgesetzt
40 printchr$(147)"eingegebenes datu
m:":fori=1ton:printn$(i)" ";:next:g
etkeya$:run
50 printchr$(19)chr$(19)chr$(27)+"1
"chr$(147):end
1000 rem ***************************
*
1001 rem ***** menue aufbauen *****
*
1002 rem ***************************
*
1010 n=3:rem anzahl d. daten
1020 tl=10:rem textlaenge ohne prom
pt
1030 rem  "text.....prompt",x,y
1040 data "Tag          20",10,5
1050 data "Monat        Mai     ",10,1
0
1060 data "Jahr         1987",10,15
1070 restore1040:printchr$(147)chr$
(27)+"m"
1080 fori=1ton:read a$(i),x(i),y(i)
:char,x(i),y(i),a$(i):x(i)=x(i)+tl
1090 a$(i)=right$(a$(i),len(a$(i))-
tl)
1100 f$(i)=left$(cr$,len(a$(i))-1):
next
1109 rem ***************************
*
1110 rem ** input-routine  ********
*
1111 rem ***************************
*
1120 gosub 1190:rem fenstereingabe
1130 gosub1160:rem werte aus fenste
r
1140 return
1150 rem * werte aus fenster lesen
**
1160 fori=1ton:char,x(i),y(i),""
1170 open3,3:fork=1tolen(a$(i)):get
#3,f$:n$(i)=n$(i)+f$:nextk:close3:n
exti:return
1180 rem ***** fenstereingabe *****
*
1190 i=1:gosub1260
```

BILDSCHIRM ALS NOTIZZETTEL

Ist Ihnen nicht auch schon einmal passiert, daß Sie

```
1200 getb$:if b$=chr$( 13) then retu
rn
1210 if b$=chr$( 145) and i>1 then i
=i-1:gosub1260:goto1200
1220 if b$=chr$( 17) and i<n then i=
i+1:gosub1260:goto1200
1230 if b$=""thengosub1280:goto1200
1240 gosub1290:printb$;:goto1200
1250 rem ***** eingabefenster *****
*
1260 gosub1290:wi$=chr$( 27)+"t"+f$(
i)+chr$( 27)+"b"+chr$( 19)
1270 char,x( i),y( i),wi$:return
1280 cp=3072+40*peek( 205)+peek( 202)
:pokecp,peek( cp)or128:return:rem re
verse
1290 cp=3072+40*peek( 205)+peek( 202)
:pokecp,peek( cp)and127:return:rem n
ormal
```

```
10 rem *****************************
20 rem text plus4 auf script/plus
30 rem *****************************
40 poke56,20:clr
50 input"filename plus4";p$:open8,8
,8,p$+",s,r":i=5120
60 get#8,a$:pokei,asc( a$):gosub150:
i=i+1:pokek,asc( a$):ifst=0then60
70 close8:hlt=i
80 input"filename script/plus";p$:o
pen8,8,8,p$+",s,w":hlt=i
90 fori=5120 to hlt
100 gosub150:print#8,d$;
110 next
120 close8:poke56,253:clr:end
130 rem * umwandlung bildschirmcode
140 rem ****** in ascii-code ******
150 bc=peek( i)and127
160 c=bc-64*( bc<32orbc>95)-128*( bc>
63andbc<96)
170 d$=chr$( c):printd$;:return
```

das im Speicher befindliche Programm versehentlich gelöscht haben? Zum Glück gibt es dafür die Möglichkeit, mit einem POKE und zwei SYS das Ganze zu retten. Aber wo steckt jetzt diese Ausgabe der COMMODORE WELT oder meine Notiz? Merken kann man sich ja auch nicht alles. Unser Tip: Legen Sie sich für die Zukunft eine Diskette (oder Kassette) mit dem Namen 'Merkzettel' an. Dann geben Sie einmal nach dem Löschen des Bildschirms ein:

POKE4097,1:SYS34840
:SYS34891:CLR:
LIST

Nun gehen Sie mit 'moN' in den Tedmon und geben ein (bei Kassette statt 8 eine 1):

S"RENEW",8,0C00,
0C50 (RETURN)
X (RETURN)

Jetzt werden die ersten beiden Bildschirmzeilen abgespeichert. Beim nächsten Unfall braucht man jetzt nur noch die

ELEKTRONISCHER MERKZETTEL

Diskette (Kassette) einzulegen und mit:

LOAD"RENEW"
,8,1 (RETURN)

(oder ',1,1') diese beiden Zeilen absolut zu laden, wodurch der Text wieder auf dem Bildschirm erscheint und mit HOME und RETURN hat man sein Programm wieder in alter Frische vorliegen. Den Trick kann man natürlich für alle Notizen anwenden. Wichtig ist, daß das abgespeicherte File einen Namen hat, der erkennen läßt, was er beinhaltet. Steht ein Programm im Speicher, so sollte man das Einladen mit dem TEDMON machen:

moN (RETURN)
L"RENEW"
,8 (RETURN)
X (RETURN)

Bei längeren Texten muß 0C50 erhöht werden.

JEDES BYTE IST KOSTBAR

Besonders der C16/116-Besitzer kennt das Problem des Platzmangels. Die 12 KByte, mit denen man ohne Grafik arbeiten kann, sind zwar in vielen Fällen ausreichend, aber sobald die Grafik eingeschaltet wird, kommt es auf jedes Byte an.
Natürlich will man seine Programme übersichtlich gestalten, daher wenige Befehle in eine Zeile packen und noch zusätzlich REM-Zeilen zur Erklärung einfügen. Auch die Leerzeichen zwischen den Befehlen sind eigentlich nicht nötig, aber man kann halt 'POKE AS,W' besser lesen als 'POKEAS,W'. Vielleicht haben Sie auch bereits ein Datenverwaltungsprogramm geschrieben, aber keinen Platz mehr, um die Daten zu speichern. Damit Sie endlich mal unter Ihren Programmen tüchtig aufräumen können, liefern wir Ihnen drei sehr nützliche Programme.
Da wäre zunächst einmal der SUPERZEILER: Mit dem C16 kann man maximal 88 Zeichen auf einmal eingeben. Der Interpreter kann aber Basiczeilen verarbeiten, die bis zu 255 Byte lang sind, wobei Befehlsworte wie PRINT, GOSUB... jeweils nur als ein Byte abgespeichert sind.
Jede Zeile braucht zusätzlich zu den eigentlichen Befehlen noch 4 weitere Byte (Linkadressen und Zeilennummer). Man kann also Speicherplatz sparen, wenn man mehrere Zeilen hintereinanderhängt. Mit diesem Programm kann man nun beliebige Zeilen miteinander verketten.
Nachdem der Superzeiler abgetippt und abgespeichert ist, lädt man das zu verändernde Programm. An den Anfang jeder Zeile, die man mit der vorhergehenden verketten will, schreibt man (in den Zwischenraum nach der Zeilennummer) einen 'Gartenzaun': #. Dann wird, wie am Ende des Programms beschrieben, der Basicanfang hochgesetzt, der Superzeiler geladen und gestartet. Nachdem das Programm fertig ist, müssen natürlich der Basicanfang und das Basicende wieder heruntergesetzt werden. Die entsprechenden Adressen für die Pokebefehle werden errechnet und auf dem Bildschirm ausgedruckt. Nachdem Sie zweimal auf Return gedrückt haben, steht das Programm in komprimierter Form im Speicher. Sie müssen bei dem Verbinden von zwei Zeilen darauf achten, daß die erste kein REM, IF... enthält, wodurch die zweite nicht zur Ausführung kommen »

kann, und daß die zweite nicht von einer anderen durch GOTO oder GOSUB angesprungen wird.

AUSSUCHEN IST MÖGLICH

Der Vorteil des Superzeilers gegenüber dem Basickompressor, welcher (für den C64) in einer der COMMODORE WELT-Ausgaben erschien, ist die Tatsache, daß man sich aussuchen kann, welche Zeilen man verknüpfen will und daß es wesentlich weniger Platz braucht, so daß das zu verändernde Programm länger sein darf.
Aber bevor Sie nun fleißig verketten und Ihr Programm mit überlangen Zeilen versehen, die dann ja leider nicht mehr veränderbar sind, sollten Sie noch alle REM-Zeilen, Leerzeichen und Doppelpunktzeilen (diese dienen zum Strukturieren des Programms und enthalten nur einen Doppelpunkt) entfernen. Dazu dient der REM-KILLER. Die Anwendung ist am Ende des Programms beschrieben.
Bitte beachten Sie, daß im Programm keine REM-Zeile mit GOTO, GOSUB... angesprungen wird. Wenn dies doch der Fall war, so merken Sie das, indem Sie den RENUMBER-Befehl eingeben. Es erscheinen dann Sprungziele nach 65536. Bei einem zweiten RENUMBER erhalten Sie einen SYNTAX ERROR wegen dieser unzulässigen Zeilennummern.
Wenn Sie nun beide Programme haben, sollten Sie diese erst einmal auf sich selbst anwenden. Sie werden staunen, wie kurz sie werden. Dadurch sparen Sie nicht nur Speicherplatz, sondern auch immens viel Ladezeit, was sich besonders bei Kassettenbetrieb bemerkbar macht.
Wenn Ihr ursprüngliches Programm schon so groß ist, daß der SUPERZEILER und der REM-KILLER keinen Platz mehr haben, so gibt es noch die Möglichkeit, das Programm mit Hilfe der DELETE-Funktion in zwei Teile zu spalten und

KEYTASTEN WERDEN NEU BELEGT

nach dem Komprimieren wieder zu verbinden.
Auch wäre es doch manchmal sinnvoll, den SUPERZEILER und den REM-KILLER zu einem Programm zu machen.
Deshalb hier nun unser drittes Hilfsprogramm: Programme verbinden mit KEY-MERGE
Um zwei Basicprogramme aneinander zu hängen, muß zuerst das eine geladen und mit neuen, niedrigen Zeilennummern versehen werden. Dann wird der Basicanfang auf das Ende des ersten Programms hochgesetzt und das zweite geladen, welches wiederum mit neuen, höheren Zeilennummern versehen werden muß.
Dann muß der Basicanfang wieder auf den Normalwert gesetzt werden. Das ganze ist recht eintönig und fehlerträchtig. Aber wozu hat man denn einen Computer, wenn man alles alleine machen muß?
Dieses Programm belegt die KEY-Tasten neu, so daß man alles mit wenigen Tastendrücken erledigen kann. Eine Anleitung steht am Ende, damit Sie sie auch noch finden, wenn Sie diese COMMODORE WELT mal verlegt haben.

```
10 rem *** key-belegung ***********
20 rem *** zum mergen ************
30 cr$=chr$( 13) :cl$=chr$( 147)
40 hk$=chr$( 34)
50 a$=",( pE( 45) +256*pE( 46) -2) "
60 fori=1to21:p$=p$+chr$( 29) :next
70 a=1375:fori=0to7:pokea+i,0:next
80 key 1,"renU1,1"+chr$( 13) +"p043"+a
$+"and255:p044"+a$+"/256"+cr$
90 key 2,"dL"+p$+":"+cr$
100 key 3,cl$+"diR"+cr$
110 key 4,"renU999,1"+cr$+"p043,1:p0
44,16"+cr$
120 a=1375:fori=0to7:b=b+peek( a+i) :n
ext:printb:end
130 rem anleitung:
140 rem 1) mit f3 directory ansehen
150 rem    mit cursor auf zeile mit
160 rem    filenamen fahren
170 rem 2) mit f2 laden
180 rem 3) f1 drucken
190 rem 4) wieder 1)+2) fuer zweites
200 rem    file
210 rem 5) f4 und fertig (oder weite
r
220 rem    2)+3) wenn noch mehr file
s)
```

PROGRAMM LÄDT SELBSTTÄTIG

Zum Programm noch ein paar Anmerkungen:
In Zeile 70 werden alle Tastenbelegungen gelöscht. Die Zeilen 90 und 100 sind nur für den Floppy-Benutzer interessant. Mit f3 wird das Directory gelistet, wobei vor dem DIRECTORY-Befehl ein CLEAR steht. Dadurch muß man nicht erst eine leere Zeile suchen, bevor man f3 drükken kann. Danach fährt man einfach auf die Zeile, in der das zu ladende Programm steht und drückt f2. Es wird an den Anfang der Zeile dL=DLOAD geschrieben und vor das PAG ein Doppelpunkt gesetzt. Nun lädt sich das Programm selbstätig und kann mit dem zweiten (mit f1 und f4) verkettet werden.
Am Schluß sollte noch ein RENUMBER eingegeben werden.
Zeile 120 berechnet den für die KEY-Belegung benötigten Platz. Von 128 Byte werden 127 gebraucht.
Nun viel Spaß beim Aufräumen.

WIE ZUFÄLLIG IST RND?

Will man ein Spiel programmieren oder eine Zufallsgrafik erschaffen, so kommt man um die sogenannte Random-Funktion RND nicht herum. Diese wird vielfach falsch angewendet, vor allem deshalb, weil das Ergebnis der RND-Funktion vom Argument abhängt. Obwohl sie im Handbuch eigentlich nicht so schlecht erklärt ist, wird sie selten richtig zweckmäßig eingesetzt.
Was ist nun eigentlich Zufall?
An eine zufällige Verteilung wird vor allem die Bedingung gestellt, daß keine Wiederholungen von Zahlenfolgen vorkommen und alle Zahlen gleich oft erscheinen. Wirklich zufällig sind physikalische Prozesse wie der radioaktive Zerfall oder das Rauschen eines elektronischen Verstärkers. Nun sollten aber die Speicherinhalte eines Computers nicht vom Zufall abhängen, sondern festen Regeln folgen. Dies macht auch die RND-Funktion. In fünf Speicherstellen (1283–1287) stehen nach dem Einschalten 5 Werte für die sogenannte Kernzahl (0,255,0,255,0). Aus diesen wird dann die Zufallszahl in Abhängigkeit vom Argument berechnet.

```
10 rem = rem-killer ================
20 rem (p)            cbm revue team=
30 rem ==============================
40 rem (c)         r. schmid-fabian=
50 rem c16/116 plus4                =
60 rem ==============================
70 ho$=chr$(19):cl$=chr$(147):scnclr
80 dim i,c,hk,lv,zn,lz,hz,ab,eb
90 nb=4096:nm=nb:rem nulladr. d. bas
iczeile vorh./nachh.
100 goto 340
110 rem ****************************
120 rem *** basicanfang und -ende **
130 rem *** und zeilennummern ******
140 rem ****************************
150 lv=nm+4:ab=nb+5:eb=peek(nb+1)+25
6*peek(nb+2)-2:hk=0
160 if eb<0 then 260
170 lz=peek(nb+3):hz=peek(nb+4)
180 zn=lz+256*hz:printho$"zn="zn
190 return
200 rem ***** neue adresse berechnen
***
210 ad=lv+2
220 ha=int(ad/256):la=ad-ha*256
230 pokenm,0:pokenm+1,la:pokenm+2,ha
240 nb=eb+1:nm=lv+1:return
250 rem ****** programm beenden*****
***
260 pokenm,0:pokenm+1,0:pokenm+2,0
270 ha=int((nm+3)/256):la=(nm+3)-ha*
256
280 printcl$"poke44,16:poke45,";:pri
ntla;:print":poke46,";:printha;
290 poke1319,19:poke1320,13:poke239,
2:end
300 rem ***************************
310 rem ** rem und blank entfernen
320 rem ** zeilennummern eintragen
330 rem ***************************
340 gosub 150:rem anfang,ende berech
n.
350 pokenm+3,lz:pokenm+4,hz:rem zeil
ennr.
360 if ab=eb and peek(ab)=58 then lv
=lv-5:goto480
370 for i=ab to eb:c=peek(i)
380 if c=34 then if hk=0 then hk=1:
goto 460
390 if c=34 then if hk=1 then hk=0:
goto 460
400 if hk=1 then 460 :rem in anfuehr
.z.
410 if c=32 then 470 :rem blank
420 if c<>143 then 460
430 if i=ab then lv=lv-5:i=eb:goto47
0
440 if peek(i-1)=58 and i>ab then lv
=lv-1:rem doppelpunkt vor rem entfer
nen
450 i=eb: goto 470
460 lv=lv+1:pokelv,c:rem kopieren
470 next i
480 if hk=1 then lv=lv+1:pokelv,34
490 gosub 210:rem neue adresse
500 goto 340
510 **** bemerkungen: *************
520 erstes programm laden
530 basicanfang hochsetzen:
540 poke44,peek(46)+1        (RETURN)
550 poke peek(44)*256,0:new(RETURN)
560 jetzt remkiller laden, starten
```

```
10 rem == superzeiler ==========c 16
20 rem (p) 08/86  commodore welt   =
30 rem ==============================
40 rem (c) 07/86 r. schmid-fabian =
50 rem c16/116 plus4               =
60 rem ==============================
70 ho$=chr$(19):cl$=chr$(147)
80 c1$=chr$(17):p1$=c1$+c1$+c1$+c1$
90 nb=4096:nm=nb:rem
nb=basicanfang=peek(44)*256
100 lv=nm+4:ab=nb+5:eb=peek(nb+1)+25
6*peek(nb+2)-2
110 if eb<0 then goto 220
120 pokenm,0:pokenm+1,1:pokenm+2,1
130 lz=peek(nb+3):pokenm+3,lz
140 hz=peek(nb+4):pokenm+4,hz
150 print"zn="lz+256*hz
160 fori=abtoeb:co=peek(i)
170 if i=ab and co=35 then lv=lv-4:p
okenm,58:goto190
180 lv=lv+1:pokelv,co
190 next
200 nb=eb+1:nm=lv+1
210 goto100
220 pokenm,0:pokenm+1,0:pokenm+2,0
230 ha=int((nm+3)/256):la=(nm+3)-ha*
256
240 printp1$"poke44,16:poke45,";:pri
ntla;:print":poke46,";:printha
250 print:print:print"63999":print"l
ist"ho$
260 end
270 rem***************************
280 bemerkungen:
290 erstes programm-laden
300 an den anfang jeder zeile, die
310 mit der vorhergehenden verkettet
320 werden soll ein '#' setzen
330 basicanfang hochsetzen:
340 poke44,peek(46)+1
350 poke peek(44)*256,0:new
360 jetzt superzeiler laden, starten
370 verkettete zeilen duerfen nicht
380 mit goto... angesprungen werden!
```

a) Argument ist negativ (A=RND(–1):
Die Kernzahl wird zurückgestellt auf einen vom Argument abhängigen Wert. Das gleiche Argument ergibt immer das gleiche Ergebnis. PRINT RND(–1) ergibt immer: 2.99196472E–08
Danach stehen in den 5 Speicherstellen die Zahlen 104,0,129,0,0. Man kann diese auch hineinpoken und erhält den gleichen Effekt.

b) Argument ist Null (A=RND(0)):
Hier wird die Kernzahl mit Hilfe der internen Uhr eingestellt. PRINT RND (0) gibt immer einen anderen Wert, selbst wenn man versucht, die gleiche Ausgangsbasis zu schaffen. Versuchen Sie einmal
A=RND(–1):TI$= "000000":PRINT RND(0)
im Direktmodus. Sie werden immer ein anderes Ergebnis erhalten. Hier scheint also wirklich Zufälligkeit vorzuliegen.

c) Argument ist positiv:
Ergibt immer neue 'Zufallszahlen', die auf der vorhergehenden Kernzahl basieren. Stellt man also mit A=RND(–1) die Kernzahl auf einen bestimmten Wert, so erhält man mit FOR I=1 TO 10:PRINT RND(1): NEXT
immer die 10 gleichen Zahlen. Der Wert des Arguments ist nicht wichtig, solange er positiv ist.

WARUM GIBT ES UNTERSCHIEDLICHEN ZUFALL?

Die eingebaute Zufallsfunktion soll dafür sorgen, daß alle Werte möglichst gleich oft auftreten. Dies ist wichtig, wenn man in einem Würfelspiel keine Zahl bevorzugen, oder mit der RND-Funktion die Gültigkeit einer mathematischen Aussage für alle Zahlen feststellen will. Damit man nun aber die Möglichkeit, hat, einen Test zu den gleichen Bedingungen zu wiederholen, ist es wichtig, daß die Folge der Zahlen so gewählt werden kann, daß sie immer gleich ist. Dazu stellt man die Kernzahl mit einem negativen Argument auf einen bestimmten Wert ein, läßt den Test laufen und kann das Ergebnis auch jederzeit reproduzieren. Nimmt man dann ein anderes negatives Argument, erhält man eine andere reproduzierbare Zahlenfolge.
Bei einem Vokabellernprogramm kann man die abzufragenden Vokabeln in ein Feld einlesen und dann mit RND auswählen. Eine andere elegante Methode ist folgende: Man schreibt jede Vokabel mit ihrer Bedeutung in eine DATA-Zeile und verwendet den RESTORE-Befehl. Dazu müssen die Zeilennummern natürlich einen definierten Abstand besitzen. Ein Beispiel: Die DATA-Zeilen gehen von 1000 bis 1300 im Abstand 10. Dann kann man mit

```
100 ZN = INT (RND(1)
    *30)*10+1000
110 RESTORE ZN:
    READ ENGL$,
    DTSCH$
```

die Werte aus einer beliebigen DATA-Zeile auslesen. Damit aber die Lernerei nicht immer bei der gleichen Vokabel beginnt, sollte die Kernzahl mit A=RND(–TI) auf einen zufälligen Wert eingestellt werden (die Zeit TI, die seit dem Einschalten vergangen ist, ist sehr zufällig).

WELCHER ZUFALL IST AM ZUFÄLLIGSTEN?

Man hat nun zwei Möglichkeiten, Zufallszahlen zu bekommen: RND(1) und RND(0). Will man entscheiden, wie zufällig diese sind, kann man sich natürlich mehrere Stunden lang Zahlenreihen betrachten und versuchen, darin eine Regelmäßigkeit zu erkennen.
Hier sind zwei wirkungsvollere Methoden:

a) Das Gartenzaunrennen:

```
10 print rnd(–ti), chr$
   (147
20 x=int(rnd(1)*10)
30 z(x)=z(x)+1:if z(x)
   =5 then z(x)=0
   :z$(x)=z$(x)"#"
40 print chr$(19)
50 for i=0 to 9:print i;
   z$(i):next
60 goto 20
```

Mit diesem kleinen Programm können Sie verfolgen, ob die Zahlen 0–9 gleich oft vorkommen. Nach dem Starten erscheinen die Zahlen untereinander und rechts davon bildet sich je ein Gartenzaun. Sie können nun mit Ihren Freunden wetten, welcher zuerst ankommt. Soll es schneller gehen, so kann man die 5 in Zeile 30 verkleinern. Will man jedoch die Statistik beurteilen, so muß man eine große Zahl wählen und es sehr lange laufen lassen. Dann ist eine Abfrage nötig, wenn ein String den rechten Rand erreicht hat:

```
50 for i=0 to 9:print i;
   z$(i)
60 if len(z$(i))Ä38 then
   next:goto 20
```

b) Die grafische Methode:
Wird zu einer Zahl im Mittel gleichviel dazugezählt wie abgezogen, so bleibt das Ergebnis immer nahe bei der Ausgangszahl. Damit können wir nun die beiden in Frage kommenden RND-Möglichkeiten testen. Das folgende Programm zeichnet die x- und y-Werte und damit kann man sehen, ob es in der Zufallsfunktion Regelmäßigkeiten gibt. Und tatsächlich gibt es die bei RND(1) sehr stark, wie der Bildschirm zeigt. Sie sehen, daß sich die gleiche Sequenz beständig wiederholt. Nun setzen Sie mal RND(0) ein und wiederholen das Ganze mit verschiedenen Startwerten für x und y.
Sie werden sich wundern. Bei manchen Werten scheint sich der kleine Flimmerpunkt sehr wohl zu fühlen, bei anderen Startwerten wandert ein Wurm quer über den Bildschirm.

```
10 graphic1,1:scale0
20 ti$="000000":a=
   rnd(–1)
30 x=60:y=100
40 x=x+rnd(1)–0.5:y=
   Y+RND(1)–0.5
50 draw 0,x,y:draw ,x,y
60 goto 40
```

Sie haben nun die Möglichkeit, die verschiedenen RND-Verfahren zu testen. Bei RND(0) ist Zustand der inneren Uhr mitentscheidend und natürlich auch die Zeitspanne zwischen den Aufrufen von RND. Wenn Sie Zufallsgrafiken machen wollen, sollten Sie jedenfalls etwas mit den Möglichkeiten spielen und zwischendurch die Kernzahl mit RND(–TI) verändern. Auch ein Verschachteln von RND ist möglich: PRINT RND(RND(0))
Viel Spaß beim Spiel mit dem Zufall.

HIRES-BILD ABSPEICHERN:

Dies geht am besten mit dem Monitor.
Um dies zu vereinfachen, hier eine Funktionstastenbelegung:

```
80 r$=chr$(13):h$=chr$
   (34)
90 p$="rnd(1)"
100
key1,"moN"+r$+"1"+h$
   +p$+h$+",8"+r$+"x"
   +r$+"gR1"+r$
110
key2,"moN"+r$+"s"
   +h$+p$+h$",8,2000,
   4000"+r$+"x"+"
   g0200"+r$
120 scnclr:graphic1,1
130 char,10,5,"bitte f1
    druecken"
140 end
200 rem programmfort
    setzung
```

Jetzt braucht man nur f2 zu drücken, und das Grafikbild wird auf die Diskette abgespeichert, mit f1 wird es wieder geladen. Bei Kassette muß man statt der 8 eine 1 schrei-

ben. Will man die Farbe mit abspeichern, setzt man für 2000 1800 ein. Der Name in pS kann beliebig gewählt werden. Die Zeilen 120–200 zeigen, wie man im Programm die Aufforderung f1 zu drücken einbinden kann.

KEY-BELEGUNG NORMALISIEREN:

Will man nach Veränderung der Funktionstastenbelegung wieder den Einschaltzustand erreichen, so kann man einen RESET bei gleichzeitigem Drücken der RUN/STOP-Taste durchführen. Man befindet sich dann im Monitor und kann mit x wieder ins Basic. Der Nachteil besteht darin, daß auch die Bildschirmfarben wieder im Ursprungszustand sind. Eine bessere Möglichkeit, die man auch vom Programm aus nutzen kann, besteht in:
SYS 62359 (Funktionstasten wieder normal)

BESSERE LESBARKEIT DER ZEICHEN:

Die beim C16/C116 gewählten Farben nach dem Einschalten sind sehr ungünstig. Man kann oft das M und das N schwer unterscheiden. Bessere Lesbarkeit erhält man mit:
coL0,2,3:coL4,2,4

FARBEN ANGLEICHEN:

Will man dem Rahmen die gleiche Farbe wie dem Hintergrund geben oder die Zeichen unsichtbar machen (zm z.B. den Monitor unsichtbar aufzurufen...) so kann man RCLR und RLUM verwenden:
COLOR 4,RCLR(0), RLUM(0) (Rahmen = Hintergrund)
COLOR 1,RCLR(0), RLUM(0) (Zeichenfarbe = Hintergrundfarbe)

LISTING RETTEN NACH NEW:

Hat man versehentlich ein NEW eingegeben oder einen RESET ausgelöst, ohne RUN/STOP zu drükken, so kann man das Listing wiederholen mit:
POKE4097,1:SYS34840 :SYS34891:CLR
Ist der BASIC-Anfang nicht bei 4096, so gibt man statt POKE4097,1 ein:
POKE PEEK(43)+256* PEEK(44),1

GRAFIK BESCHRIFTEN MIT 64 ZEICHEN JE ZEILE

Mit dem CHAR-Befehl des C16/116 kann man zwar Texte und Grafikzeichen in den HIRES-Bildschirm schreiben, aber er hat zwei entscheidende Nachteile: Er kann die Zeichen nicht an beliebigen Orten darstellen und es sind keine Kleinbuchstaben möglich.
Dieses Programm schafft Abhilfe.
In den Data-Zeilen steht ein neuer Zeichensatz. Diese Zeichen sind nicht 8 Bit (Punkte) breit, sondern nur 4. Mit je einem Punkt Abstand kann man so 64 Zeichen in einer Zeile unterbringen. Der Trick besteht darin, die Zeichen aus den DATA-Zeilen direkt in einen String einzulesen, der dann mit GSHAPE dargestellt wird. Der Zeichensatz umfaßt alle Zeichen mit CHR$-Codes von 32–127 (s. Handbuch S. 215).
Damit das Programm nun nicht abstürzt, wenn ein falsches Zeichen gedrückt wird, werden die gültigen Zeichen in Zeile 130 in g$ zusammengefaßt und mit der INSTR-Funktion die Lage des Zeichens in g$überprüft (Zeile 180). Ist es nicht vorhanden, erhält man Null und das Programm überprüft nun, ob eine der Cursortasten gedrückt wurde. Mit denen kann man nämlich einen Strichcursor an jede beliebige Stelle des Bildschirms bringen und dort einfach losschreiben. Mit der RESTORE-Funktion wird die DATA-Zeile ausgewählt, die dem Zeichen entspricht und mit der MID$-Funktion werden die entsprechenden Bytes in s$ ersetzt.
Beim 'Drucken" des Zeichens mit GSHAPE wurde die EOR-Verknüpfung gewählt (,4). Daher wird automatisch auf hellem Untergrund dunkel und auf dunklem hell geschrieben. Mit RETURN springt der Cursor an den Anfang der Folgezeile und kann von da mit den Cursortasten positioniert werden. Will man die Buchstaben mit größerem Abstand darstellen, so verändert man d in Zeile 120.

PLATZPROBLEME BEIM C16

Obwohl das eigentliche Programm klein ist, kann das Programm auf dem C16 ohne Erweiterung nicht in dieser Form eingesetzt werden, da die DATA-Zeilen zuviel Platz brauchen. Man kann sich jedoch einen Teil der Zeilen auswählen und die restlichen DATA-Zeilen löschen.
Hierbei entspricht Zeile 301 CHR$(32), 302 CHR$(33)... Das Schreiben geht schnell genug, so daß man das Programm auch für ein eigenes Dateiverwaltungsprogramm oder Textprogramm verwenden kann.
Man kann es auch beliebig abändern, indem man zum Beispiel den Zeichensatz ändert und die hier angewandte Methode benutzt, um aus selbstdefinierten Symbolen Bilder zu machen. Oder Sie können die Zeichen auch größer machen und damit komplizierte mathematische oder chemische Formeln schreiben.
Wer das Sonderheft über den C16 besitzt, kennt auch die Möglichkeit, den Zeichensatz zu verändern. Dieser Minizeichensatz läßt sich in Spielen effektvoll einsetzen.

MERGEN VON KLEINEN PROGRAMMTEILEN:

Will man kleine Teile eines Programms in ein anderes übernehmen, listet man diese Zeilen auf dem Bildschirm, lädt das zweite Programm in den Speicher und übernimmt dann die gelisteten Zeilen mit RETURN. Aber Vorsicht! Wenden Sie dabei nicht den Fensterbefehl an.

FENSTER ZERSTÖRT BILDSCHIRMAUFBAU:

Hat man ein paar Zeilen gelistet, die länger als 40 Zeichen sind und gibt dann ESC&T oder ESC&B ein, um ein Fenster zu bilden und löst dieses danach wieder mit zweimal HOME auf, so sind alle Programm-Zeilen, die im Fenster waren, nur noch einzeilig. Dies kann auch gewünscht sein, wenn man aus langen Zeilen kurze machen will, um noch etwas einzufügen.

SPEICHERPLATZ SPAREN IN DATA-ZEILEN:

Hat man Daten für Maschinenprogramme, SHAPES oder Zeichensatz in DATA-Zeilen abgelegt, so stehen dort Zahlen, die kleiner 256 sind. Mit Hilfe des HEX-Befehls kann man diese in einen String verwandeln: Bei einer DATA-Zeile mit 8 dreistelligen Zahlen spart man so 16 Bytes. Bei vielen DATA-Zeilen macht sich dies stark bemerkbar. Zum Beispiel spart man beim 64-Zeichen-Programm aus der 9/86 ganze 750 Bytes. Folgendes Programm übernimmt die Umwandlung und die letzte Zeile die Rückwandlung in Zahlen.
10 rem * hexdata verwandelt dezimal

```
20 rem ** daten in hex-
datenstring
50000 read a:zn=peek
(64)+256*peek(64)
50010 restore zn
50020 a$=mid$(str$(zn),
2)+"dA":fori=1to8:reada
:a$=a$+right$(hex$(a),
2):next
50030 printchr$(147)a$
:print"zn="zn+10":go
50010"
50040 t=1319:poket,19
:poket+1,13:poket+2,13
'poke239,3:end
50050 data 100,110,120,
130,140,150,160,170
50060 data 100,110,120,
130,140,150,160,170
50070 data 100,110,120,
130,140,150,160,170
50080 data 100,110,120,
130,140,150,160,170
50090 data 100,110,120,
130,140,150,160,170
50100 rem * rueckrech-
nungsbeispiel
50110 reada$:fori=1to8*2
step2:b=dec(mid$(a$,i,2))
:printb;:next
```

BASIC SCHNELLER DURCH BILDSCHIRM ABSCHALTEN:

Mit POKE 65286,11 kann man den Bildschirm abschalten. Dadurch kann der Computer die für den Bildaufbau nötige Zeit für Berechnungen oder das Erstellen von Grafiken verwenden. Die Grafik kann dann natürlich erst nach dem Wiedereinschalten mit POKE 65286,27 sichtbar werden. Bei der Programmentwicklung diesen POKE auf eine Funktionstaste legen, damit man bei Programmabsturz die Fehlermeldung lesen kann. Dieses kleine Programm zeigt den Zeitunterschied für eine Berechnung mit und ohne Bildschirmabschalten. Es geht 33% schneller.

```
10 ti$="000000":poke
65286,11:fori=1to200
:a=sin(i*i):next:poke
65286,27:printti
20 ti$="000000":poke
65286,27:fori=1to200
:a=sin(i*i):next:poke
65286,27:printti
```


IMPRESSUM

COMMODORE-SPECIAL

erscheint achtmal jährlich (4x C16, 4x 128) in der CA-Verlags GmbH (i.G.)

VERANTWORTLICH FÜR DEN INHALT:
Alfons Mittelmeyer

REDAKTION UND STÄNDIGE MITARBEITER:
Peter Basch, Harald Beiler, Rosemarie Huber, Lothar Miedel, Michael Reppisch, Rudolf Schmid-Fabian, Torsten Seibt, Hermann Wellesen, Bernd Welte

GESCHÄFTSFÜHRER (und verantwortlich für Anzeigen):
Werner E. Seibt

ANSCHRIFT FÜR ALLE VERANTWORTLICHEN:
Postfach 1107,
8044 Unterschleißheim
Tel.: 089/129 80 11
Telex: 5214428 cav-d
Es gilt Preisliste Nr. 7 vom 1.2.1987
Media-Unterlagen bitte anfordern.

VERTRIEB:
Verlagsunion Wiesbaden